行政院國家科學委員會專題研究計畫　成果報告

序的最佳化方法的改進(2/2)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-009-027-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立交通大學電機與控制工程研究所

計畫主持人：林心宇

計畫參與人員：洪士程，黃榮壽，張紹興，陳亮元，林啟新.

報告類型：完整報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 10 月 12 日

國科會研究計劃成果期末報告

計劃名稱：序的最佳化方法的改進(2/2)

計劃執行起迄：2003.08.01 至 2004.07.31

計劃主持人：林心宇

計劃編號：NSC-92-2213-E-009-027

執行機關：國立交通大學電機與控制工程研究所

計畫中文摘要

哈佛大學何毓琦教授於 91 年元月初在交大開設為期三天的短期課程，其中一個最重要的主題 Ordinal Optimization 引起參與此課程的各校教授熱烈的迴響，主因是 Ordinal Optimization 是一個應用範圍非常廣的研究主題，而且它對一些以往被認為不能解的最佳化問題提供了一個相當不錯的解答。

鑑於 Ordinal Optimization 應用範圍的廣大，本研究計劃針對現有的 Ordinal Optimization 的方法做更進一步的改進。我們所擬定的改進目標有二：(i)如何選取較好的 N 個樣本解。(ii)提供一個新的解 Constrained Ordinal Optimization 問題的方法。

我們已於第一年針對第(i)個主題提出解決的方法。以結合類神經網路及基因演算法來找尋較好的 N 個樣本解，並進而求得較目前的 Ordinal Optimization 方法更好的解。此研究成果已在德國慕尼黑召開 2003 ASMC 會議中發表。而在本年(即第二年計劃)中，我們針對 Constrained Ordinal Optimization 的問題提出一個具體可行的解決方法及其理論上的支持，並將此方法應用在解具離散控制變數的最佳電力潮流問題上。

Abstract

In this two-year project, we intend to propose methods for the following two significant subjects in Ordinal Optimization Theory: (i) Finding the better N samples and (ii) Constrained Ordinal Optimization Problems.

In the first subject, we have proposed a method combined neural network and genetic algorithm to find the better N samples and then to find a good enough solution, which will be better than that can be found in the current Ordinal Optimization method. This research result had been presented in 2003 ASMC held in Munich, Germany. In this year, we propose a new method to solve constrained ordinal optimization problem and apply it to the optimal power flow problem with discrete control variable.

一、前言

序的最佳化方法 (Ordinal Optimization) 是近年來由哈佛大學何毓琦教授所提出的一個解困難最佳化問題 (hard optimization problem) 的快速且有效的方法。由於這個方法的提出使得一些向來被認為無法解的 NP hard 的問題可以重新檢視其可解性。我們於上一年度(即此二年期計畫之第一年)計畫中已研擬出一個結合類神經網路及基因演算法的方法來找出巨大解集中 N 個較好的解，此結果已在德國慕尼黑舉行的 2003 高等半導體製造會議中發表[1]。而本年度我們針對有限制式的困難最佳化問題如具有離散控制變數的最佳電力潮流問題，提出一個序的最佳化演算法則。

二、研究目的

由於目前序的最佳化方法仍有一些可以改進的地方，例如：1.它所謂的 top 5% 的解在具有龐大的樣本解空間的問題中，這樣求得的解在實際最佳化技巧運用者的眼光中稱不上是很好的解。以及 2.它還不能解有限制式的困難最佳化問題，尤其是具有等式限制式的。所以我們這個計畫的研究目的，便是要克服序的最佳化方法的這兩個缺點。

我們在第一年(去年度)的研究計劃是針對第 1 個缺點來尋求改進的方法，亦即提出一個尋找較好的 N 個樣本解的方法。並應用在半導體製程的晶圓測試程序中如何訂定 lot, wafer, 及 bin 的門檻值 (threshold value) 以減少誤宰 (overkills) 及重測 (retests) 的問題。而本年度，我們希望能以極短的計算時間來求得有限制式的困難最佳化問題的一個不錯的解。我們將所研擬出的方法直接應用在具有極大挑戰性的具離散控制變數的最佳電力潮流問題上。

三、文獻探討

困難的最佳化問題在實際的世界中隨處可見，例如有離散變數的最佳化問題 (optimization problem with discrete control variables) 中的最佳化電力潮流 (optimal power flow) 問題[2-4]是最為電力工程界的研究人員所熟悉。又如隨機最佳化問題 (stochastic optimization problem)[5] 更是在離散事件中動態系統 (discrete event dynamic systems)[6] 中俯拾即是的問題。近年來，此類型最佳化問題，概皆以 simulated annealing 法[7]，基因演算法[8]，或 tabu search 法[9] 來求解，但這些方法的計算皆非常耗時而不適合實際應用。不同於這些以全域搜尋技術 (global searching technique) 來解困難最佳化問題的方法，序的最佳化方法提出了一個可迅速地求得 top n% 解的方法。何教授及其研究團隊提出了相當多的研究論文[10-11]來闡明此方法的優點：即它不像一般的 heuristic 法，它是一個可以提供 quantitative result 的方法。整個序的最佳化方法的演算步驟的理論基礎是在 Universal Alignment Probability [11]這篇文章中, 它提供了如何評量所提出的 rough model 的 roughness 以及如何選取 Selected subset 的樣本數，這些在實際的應用中非常地有用，有了這篇文章的基本概念後，對實際應用問題的序的最佳化方法的推導可說是事半功倍。

四、研究方法

我們已於上一年度計畫中針對第(i)個目標挑選了一個非常具有應用價值的晶圓測試程序中如何訂定關鍵值 (threshold values) 以減少晶粒誤宰 (overkills) 及重測 (retests) 次數的問題。在這個問題中，我們結合了類神經網路及基因演算法來做為我們在超過 10^{30} 個樣本解中挑出較好的 N (=1000) 個樣本解的方法。此晶圓測試程序中關鍵值訂定的研究成果已於 2003 年在德國慕尼黑舉行

的高等半導體製造會議中發表[1]。

本年度之計畫主要是針對第(ii)個目標，並以具離散控制變數的最佳電力潮流問題做為我們的例子。

我們針對最佳電力潮流問題所提出的改進序的最佳化方法是首先設計一個較佳的選擇 N 個樣本解的方法，並從此 N 個樣本解中以序的最佳化方法來求得一個不錯的解。此方法可簡述如下：

Step 1: 先將所有離散控制變數視為連續控制變數後，求解連續性 (continuous) 的最佳電力潮流問題。

Step 2: 根據各個離散控制變數的敏感度分析 (sensitivity analysis) 將對目標函數較不敏感的離散控制變數固定在最靠近其連續值解的離散值，僅讓最敏感的前 10 個離散控制變數可彈性設為最靠近其連續值解之左邊或右邊之離散值。

Step 3: 根據 Step 2 所選出之 10 個離散控制變數所形成的 2^{10}=1024 個組合中，根據敏感度分析，挑選出前 50 名可能具有較低之目標函數的組合。

Step 4: 將離散變數分別固定在 Step 3 所選出之組合，然後解最佳電力潮流問題。於是在這 50 個最電力潮流問題中具有最低目標函數所對應的離散控制變數的組合即是我們要找的解。

五、結果與討論

此計畫總共執行了兩年，我們已有相當豐碩的成果。計畫的第一個年度中，我們在晶圓測試的訂定關鍵值以減少誤宰與重測問題上，也獲致很好的結果，以下為我們所得結果的大略描述：

我們以實際半導體製造公司中晶圓製造的數據來建立我們隨機模擬中 Poisson distribution 的模型，且以該公司之晶圓測試程序當作我們模擬的測試程序，根據我們在研究方法中所列出之步驟，我們求得減少誤宰及重測之關鍵值。我們同時將所求得之關鍵值與 1000 個隨機產生之關鍵值比較所對應之誤宰及重測，我們所得的值是所有的關鍵值中最好的。

此結果中所使用之數據係來自某大半導體製造公司及工研院朋友們的協助，我們已在去年 3 月 31 日在德國慕尼黑召開的高等半導體製造會議(ASMC 2003)中發表，其中我們也註明了對國科會贊助本計劃研究的謝辭。

計畫的第二個年度即本年度，我們已成功地提出了一個以序的最佳化方法為基礎的解具有離散變數的最佳電力潮流問題的演算法則。此結果可概述如下：

與傳統的解法比較，我們的方法在目標函數值上平均降低 28%；同時，傳統的方法在很多情況下，無法得到可行解，而我們的方法每一次皆可得到可行解。與全域搜尋技巧的 tabu search 法比較，當 tabu search 法花了較我們的方法將近 100 倍的 CPU times 時，其目標函數值仍高於我們所得的目標函數值 18%。由此可見，我們的方法不但快速且可有效地得到一個不錯的解。這個創新的解具離散控制變數的最佳電力潮流問題的方法, 我們已於 2004 年 2 月發表於 IEEE Trans. On Power Systems[12]。為了便於參考起見，僅將論文之原稿列為本年度報告之附件。

六、參考文獻

[1] S. C. Horng, S. Y. Lin, M. H. Cheng, F. Y. Yang, C. H. Liu, W. Y. Lee and C. H. Tsai, "Reducing the Overkills and Retests in Wafer Testing Process," *2003 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, pp.286-291, Munich,

Germany, March 31~April 1, 2003.

[2] Liu, W., Papalexopoulos, A. and Tinney W., ``Discrete shunt controls in a Newton optimal power flow,"*IEEE Trans. on Power Systems*, vol. 7, no. 4, pp. 1509-1520, Nov. 1992.

[3] Tinney, W., Bright, J., Demaree, K. and Hughes, B., ``Some deficiencies in optimal power flow,"*IEEE PICA Conf. Proc.*, pp. 164-169, Montreal, May 1987.

[4] Bakirtzis, A. and Meliopoulos, A., ``Incorporation of switching operation in power system corrective control computations," *IEEE Trans. on Power Systems*, vol. 2, no. 3, pp. 669-676, Aug.1987.

[5] Ho, Y.C., Cassandras, C.C., Chen, C.H., and Dai, L., ``Ordinal optimization and simulation," *Journal of Operation Research Society*, vol. 21, pp. 490-500, 2000.

[6] Cassandras, C. G. and Lafortuene, S., Introduction to discrete event systems, Kluner Academic Pull., Bostor, MA., 1999.

[7] Chen, L., Suzuki, H. and Katou, K., ``Mean field theory for optimal power flow," *IEEE Trans. on Power Systems*, vol. 12, no. 4, Nov. 1997.

[8] Ma, J. T. and Lai, L. L., ``Evolutionary programming approach to reactive power planning,"*IEEE Proc., Generation, Transmission and Distribution*, vol. 143, no. 4, 1996.

[9] Kulworawanichpong, T. and Sujitjorn, S., ``Optimal power flow using tabu search," *IEEE Power Engineering Review*, pp. 37-40, June 2002.

[10] Ho, Y.C., ``Soft optimization for hard problem," Lecture notes, Harvard University, Cambridge, MA., 1996.

[11] Lau, T.W.E. and Ho, Y.C., ``Universal alignment probability and subset selection for ordinal optimization," *JOTA*, vol. 39, no. 3, June 1997.

[12] Shin-Yeu Lin, Yu-Chi Ho, Ch'i-Hsin Lin, “An ordinal optimization theory-based algorithm for solving the optimal power flow problem with discrete control variables,” *IEEE Trans. on Power Systems*, vol. 19, no. 1, pp. 276-286,Feb. 2004.

七、計劃成果自評

本計劃總共執行兩年，所獲成果堪稱豐碩，總計已發表了一篇知名國際會議論文，以及一篇知名期刊之論文。

附件一

An Ordinal Optimization Theory-Based Algorithm for Solving the Optimal Power Flow Problem With Discrete Control Variables

Shin-Yeu Lin, Yu-Chi Ho, and Ch'i-Hsin Lin

Abstract—**The optimal power flow (OPF) problem with discrete control variables is an NP-hard problem in its exact formulation. To cope with the immense computational-difficulty of this problem, we propose an ordinal optimization theory-based algorithm to solve for a good enough solution with high probability. Aiming for hard optimization problems, the ordinal optimization theory, in contrast to heuristic methods, guarantee to provide a top n% solution among all with probability more than 0.95. The approach of our ordinal optimization theory-based algorithm consists of three stages. First, select heuristically a large set of candidate solutions. Then, use a simplified model to select a subset of most promising solutions. Finally, evaluate the candidate promising-solutions of the reduced subset using the exact model. We have demonstrated the computational efficiency of our algorithm and the quality of the obtained solution by comparing with the competing methods and the conventional approach through simulations.**

Index Terms—**Discrete control variables, nonlinear programming, optimal power flow, ordinal optimization.**

NOMENCLATURE

u_d	n_d-dimensional vector of discrete control variables such as switching shunt capacitor banks and transformer taps.
x	n_x-dimensional vector of continuous variables consisting of real and reactive power generation, real and imaginary parts of bus complex voltage.
U_d	Sample space of u_d.
$g(x, u_d) = 0$	Real and reactive power flow balance equations.
$h(x) \le 0$	Inequality constraints such as thermal-limit security constraints, security limits on voltage magnitude, and real and reactive power generation limits.
$f(x, u_d)$	Objective function such as the total power generation cost or the total system losses.
MDCP	Mixed discrete-continuous nonlinear optimization problem such as the optimal power flow (OPF) problem with discrete control variables.
MDCP(u_d)	mixed discrete-continuous nonlinear optimization problem (MDCP) for a given u_d.
$f^o(\cdot)$	Representation of a typical objective function in ordinal optimization theory.
u_c	Continuous version of u_d.
$\underline{u}_c$, $\overline{u}_c$	Lower and upper limit of u_d.
u_{cj}, u_{dj}	The j*th* component of u_c and u_d.
CP	Continuous nonlinear optimization Problem formed by replacing u_d in MDCP with u_c.
$(x^*(u_c^*), u_c^*)$	Local optimal solution of continuous nonlinear optimization problem (CP).
u_{cj}^*	The j*th* component of u_c^*.
$\lceil u_{cj}^* \rceil$, ($\lfloor u_{cj}^* \rfloor$)	Nearest discrete value of u_{dj} to u_{cj}^* on the right-hand (left-hand) side of u_{cj}^*.
Δu_{cj}^*	$= u_{cj}^* - \lfloor u_{cj}^* \rfloor$ or $u_{cj}^* - \lceil u_{cj}^* \rceil$.
$\Delta f(\Delta u_{cj}^*)$	Deviation of optimal objective value caused by Δu_{cj}^*.
λ	Lagrange multiplier associated with $g(x, u_d) = 0$ at the optimal solution $(x^*(u_c^*), u_c^*)$ of CP.
$\nabla_{u_{cj}} g(x^*(u_c^*), u_c^*)$	Partial derivative of $g(x^*(u_c), u_c)$ w.r.t. u_{cj} evaluated at $(x^*(u_c^*), u_c^*)$.
$\nabla_{u_c} g(x^*(u_c^*), u_c^*)$	Partial derivative of $g(x^*(u_c), u_c)$ w.r.t. u_c evaluated at $(x^*(u_c^*), u_c^*)$.
$\Delta f(u_d(i))$	Deviation of optimal objective value of CP due to the change of u_c from u_c^* to $u_d(i)$.
$\mathbf{N}$	Representative set of U_d.
N	Number of samples in $\mathbf{N}$.
SS	Selected subset or estimated good enough subset which consists of the estimated top n% samples of $\mathbf{N}$.
s	Number of samples in SS.
$\tilde{x}(u_d(i))$	Approximate solution of MDCP($u_d(i)$).
k^{th}	Number of top samples in SS ordered by the objective values of the quadratic approximate solutions of MDCP(u_d) that are guaranteed to consist of a good enough solution of MDCP.

Manuscript received March 11, 2003. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant NSC91-2213-E-009-070, in part by the U.S. Army Contract DAAL03-92-G-0115 and Contract DAAH04-95-0148, in part by the Air Force under Grant F49620-98-1-0387, and by the Electric Power Research Institute under Contract WO8333-03.

S.-Y. Lin is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: sylin@ee.nctu.edu.tw).

Y. C. Ho is with the Division of Engineering and Applied Sciences, Harvard University, Cambridge MA 02138 USA.

C.-H. Lin is with the Department of Electronics Engineering, Kao Yuan Institute of Technology, Kaohsiung, Taiwan 700, R.O.C.

Digital Object Identifier 10.1109/TPWRS.2003.818732

I. Introduction

THE OPF problem has a long history in power-system research. Numerous numerical techniques had been developed for this problem such as the successive linear programming method [21]–[23], successive quadratic programming (SQP) method [4], [6], [26], Lagrange Newton method [19], [24], [25], the interior point method [28]–[30], and the recent dual-type method [11], [12], [14]. However, these methods are designed for purely continuous-variable OPF. In reality, the power systems consist of several discrete control variables such as the switching shunt capacitor banks, which are switched on and off in order to correct the voltage profile and reduce active power transmission losses and transformer taps, which are adjusted step by step to ensure that a voltage-controlled bus maintains its voltage within acceptable limits. In most of the existing OPF algorithms including the above-mentioned ones, discrete controls are treated as continuous variables until they are approximately optimized. Then they are rounded off to their nearest discrete values. Simply rounding off discrete controls can cause a considerable increase of the objective value and/or violations of inequality constraints. This deficiency had been recognized by Tinney *et al.* in [27]. A linear programming-based method was developed in [2] to deal with this type of mixed discrete-continuous nonlinear programming problems; however, it is too time consuming.

To cope with the computational intractability and the disadvantages induced by arbitrarily rounding off technique, a penalized discretization algorithm is proposed by Liu *et al.* in [15]. They employed a complicated six rules to introduce the quadratic limit penalty of a discrete control during the solution process of a Newton OPF. They finally fix the penalized continuous discrete-control at its discrete value based on a local convergence criteria. In other words, they proposed a rounding off technique based on a penalized discretization. Thus, this method cannot completely resolve the problem of infeasibility caused by not treating the discrete control variables in their exact form. To solve the OPF with discrete control variables in a more exact manner, Bakirtzis *et al.* proposed an enhanced genetic algorithm (GA) in [1], which needs only the power flow solutions for fitness evaluation, however, sacrificing the hard restriction on branch flow limits. Thus, violations of inequality constraints may occur in this method. Recent methods that treat the discrete control variables in their exact form and take the feasibility of OPF into account are the mean field theory-based annealing algorithm [5], the evolutionary algorithm [17], and the tabu search method [9]. However, these methods use global searching technique, which is very computationally time-consuming provided that the size of the search space is huge.

To retain the merits of the above three methods in handling the discrete control variables and the feasibility of OPF while avoiding their immense computational-complexity, we intend to use the ordinal optimization (OO) technique, which is recently developed by Ho and his colleagues [7], [8]. This optimization technique can effectively reduce the number of search samples of the huge sample space formed by all discrete control variables and seek a **good enough** solution with **high probability** instead of searching the best for sure. Thus, the *purpose* of this paper is to propose an OO theory-based algorithm to solve for a good enough solution of the OPF with discrete control variables efficiently. The approach of our algorithm consists of three stages. First, select heuristically a large set of candidate solutions. Then, use a simplified model to select a subset of most promising solutions. Finally, evaluate the candidate promising-solutions of the reduced subset using the exact model.

Since OO is a rather new optimization technique, we will include some relevant materials in the appendices for easier reference. Thus, our paper is organized in the following manner. In Section II, a mathematical formulation of the OPF problem with discrete control variables will be presented. In Section III, we will present our approach based on the OO theory to solve for a **good enough** solution of the OPF problem with discrete control variables. In order not to affect the fluency of presentation, a detailed review of the OO theory [7], [8], the applications of the alignment probability [10] to our problem and a comment for addressing the question regarding infeasible solutions are given in Appendices A, B, and C, respectively. In addition, an efficient dual-type method for solving a set of quadratic programming problems, which are induced from our approach as stated in Section III, is presented in Appendix D. We concluded Section III by presenting an online algorithm for obtaining a **good enough** solution of the OPF problem with discrete control variables. The test results and the comparisons of our online algorithm with the conventional approach and the competing methods on the IEEE 118-bus system and the IEEE 244-bus system are presented in Section IV. Finally, we make a conclusion in Section V.

II. Mathematical Formulation of the OPF Problem With Discrete Control Variables

The OPF problem with discrete control variables is a type of MDCP, which can be formulated as

$$\begin{aligned} \text{(MDCP)} \quad & \min_{x,u_d} f(x,u_d) \\ \text{subject to} \quad & g(x,u_d)=0 \\ & h(x)\le 0 \\ & u_d \in U_d. \end{aligned} \tag{1}$$

This optimization problem is to find the continuous and discrete control settings so as to minimize the objective function while satisfying the required constraints.

We can rewrite the MDCP shown in (1) as

$$\min_{u_d\in U_d}\{\min_{x} f(x,u_d)\,|\,g(x,u_d)=0, h(x)\le 0\} \tag{2}$$

where we denote the optimization problem inside the bracket for a given $u_d \in U_d$ as MDCP(u_d), that is

$$\text{MDCP}(u_d) \equiv \{\min_{x} f(x,u_d)\,|\,g(x,u_d)=0, h(x)\le 0\}. \tag{3}$$

In convention and practical applications of power systems, a local solution of the OPF is usually sought. However, in the MDCP shown in (2), we can tell that even if we require a local solution only, we still need to solve the MDCP(u_d) for all samples $u_d \in U_d$. Thus, suppose each discrete control variable has p possible discrete values, then there will be $p^{|u_d|}$ samples in the

sample space U_d. To give a flavor of the numerical value of the size of U_d, assuming $n_d = 50$, and $p = 4$, there will be $4^{50} \approx 10^{30}$ samples in U_d. Therefore, it will be computationally intractable to solve for a local optimal solution of MDCP shown in (2) using a global searching technique.

III. OO Theory-Based Approach

Denoting the optimal objective value of the optimization problem MDCP(u_d) for a given u_d as a function of u_d, say $f'(u_d)$, then (2) becomes $\min_{u_d \in U_d} f'(u_d)$ which has exactly the same form as the optimization problem treated in OO theory [7], [8]. Before stating our approach for the MDCP in (2), we will briefly state the idea of OO theory in the following while the details are given in Appendix A.

A. Review of OO Theory

The OO theory is a new methodology designed to cope with hard problems such as the lack of structure problems, problems with uncertainties, or problems with huge sample space that grows exponentially with respect to the problem size. The problem considered in this paper is of the latter kind. There are two basic tenets of the OO theory. The first is that of order versus value in decision making. It is obvious that determining whether $f'(u_{d1}) < f'(u_{d2})$ is much easier than determining $f'(u_{d2}) - f'(u_{d1}) =?$: consider the intuitive example of determining which of the two boxes in two hands is heavier versus identifying how much heavier one is than the other. The second tenet is the goal softening. Instead of asking the **best for sure** in optimization, it settles for the **good enough** with **high probability**. A conclusion drawn from the OO theory is the following.

Suppose we simultaneously evaluate a large set of alternatives very approximately and order them according to the approximate evaluation. Then there is high probability that we can find the actual good alternatives if we limit ourselves to the top $n\%$ of the observed good choices.

Thus, first, we use only a very rough model to **order** the goodness of a solution relying on the robustness of ORDER against noise and model error to separate the good from the bad. Second, we soften the goal of the problem and look for a **good enough** solution, which is among the top $n\%$ of the search space U_d, with **high probability**. These two steps greatly reduce the computational burden and search difficulties of the problem as will be detailed in Appendix A. A summary of these search procedures for obtaining a **good enough** solution of $\min_{u_d \in U_d} f'(u_d)$ with **high probability** can be described in the following: i) Using either a uniform selection or a heuristic method to select a representative set $\mathbf{N}$ with size N, say 1000, for the search space U_d. ii) Using an easily computed crude model to roughly evaluate and order the performance of each sample in $\mathbf{N}$ and collect the top s samples to form a selected subset (SS), which is the **estimated good enough subset**. The OO theory guarantees that SS consists of **actual good enough** solutions with **high probability**. The value of s in our approach determined based on the alignment probability [10] is 50 as will be described in Appendix A.II. iii) Evaluating the objective value of the MDCP(u_d) for each sample u_d in SS to obtain the **good enough** solution.

B. Three-Stage Approach

Based on the above search procedures, our three-stage approach for obtaining a **good enough** solution of the MDCP shown in (2) is presented in the following.

i) Using a heuristic method to determine the set $\mathbf{N}$.

First, we define the MDCP shown in (2) as a continuous nonlinear optimization problem (CP) if we replace the discrete u_d as continuous u_c. Thus, the resulting CP is shown in (4)

$$\min_{u_l \le u_c \le u_h} \{\min_{x} f(x, u_c) \mid g(x, u_c) = 0, h(x) \le 0\}. \quad (4)$$

Our strategy to determine the set $\mathbf{N}$ is to solve the CP to obtain an optimal solution $(x^*(u_c^*), u_c^*)$ first. Then, we can set each component of the discrete u_d, say u_{dj}, be either $\lfloor u_{cj}^* \rfloor$ or $\lceil u_{cj}^* \rceil$. At this stage, we have reduced the search space from p^{n_d} to 2^{n_d}. In fact, this is a similar intuition as the conventional approach for solving the MDCP in (2), because we believe that good solutions should be among the 2^{n_d} samples. However, arbitrarily rounding off does not guarantee a **good enough** solution with **high probability**. In general $2^{n_d} \gg N$, we need to reduce the search samples further. To do so, we will estimate the deviation of the optimal objective value $\Delta f(\Delta u_{cj}^*)$ caused by Δu_{cj}^* that is $\Delta f(\Delta u_{cj}^*) \equiv f(x^*(u_c'), u_c') - f(x^*(u_c^*), u_c^*)$, where $u_c' = [u_{c1}', \ldots, u_{cn_d}']^T$ in which $u_{ci}' = u_{ci}^*$, if $i \neq j$, and $u_{cj}' = \lfloor u_{cj}^* \rfloor or \lceil u_{cj}^* \rceil$. Based on the sensitivity theorem in [16], we can obtain

$$\Delta f(\Delta u_{cj}^*) \approx \lambda^T \nabla_{u_{cj}} g(x^*(u_c^*), u_c^*) \Delta u_{cj}^*. \quad (5)$$

Now if $|\Delta f(\Delta u_{cj}^*)| < \varepsilon$, a predetermined small positive real number, then we fix the discrete u_{dj} at $\lfloor u_{cj}^* \rfloor$ *if* $\Delta u_{cj}^* = u_{cj}^* - \lfloor u_{cj}^* \rfloor$ *or* $\lceil u_{cj}^* \rceil$ *if* $\Delta u_{cj}^* = u_{cj}^* - \lceil u_{cj}^* \rceil$. In other words, if u_{cj}^* is set to be $\lfloor u_{cj}^* \rfloor$ (*or* $\lceil u_{cj}^* \rceil$) and the estimated deviation of the objective value is small, then we fix u_{cj}^* at $\lfloor u_{cj}^* \rfloor$ (*or* $\lceil u_{cj}^* \rceil$). Suppose there are ℓ u_{dj} fixed by this manner, there will be $n_d - \ell$ components of u_d not yet fixed. Thus, we have further reduced the search samples from 2^{n_d} to $2^{n_d - \ell}$. The value ε is selected so that $2^{n_d - \ell} \approx N$. The above process constitutes our heuristic method for determining the set $\mathbf{N}$.

ii) Determining the selected subset (SS) based on a crude model.

Now let us denote the N samples in $\mathbf{N}$ as $u_d(i), i = 1, \ldots, N$. To pick out the samples to form SS, we will employ a crude model, which estimates the deviation of the optimal objective value in (4), that is $\Delta f(u_d(i)) \equiv f(x^*(u_c^*), u_c^*) - f(x^*(u_d(i)), u_d(i))$, due to the change of u_c from u_c^* to $u_d(i)$. The formula for estimating $\Delta f(u_d(i))$ is an extension of the formula of (5) by considering the vector increment $\Delta u_c^* = u_c^* - u_d(i)$ rather than the component increment Δu_{cj}^* and is stated in the following:

$$\Delta f(u_d(i)) \approx \lambda^T \nabla_{u_c} g(x^*(u_c^*), u_c^*)(u_c^* - u_d(i)). \quad (6)$$

We order the samples in $\mathbf{N}$ based on $|\Delta f(u_d(i))|$ obtained from (6) as follows. The sample $u_d(i)$ with lower value of $|\Delta f(u_d(i))|$, that is the sample being less sensitive to the optimal objective value of (4), will be ranked higher. In other words, the samples, which are likely to retain the optimal objective value of (4) are ranked higher. Consequently, the top ranked s $u_d(i)$ form SS, where s denotes the size of SS determined based on the alignment probability [10] as will be detailed in Appendix I. The SS thus formed is the **estimated good enough** subset. Now according to the OO theory [7], [8], SS consists of **actual good enough** samples with **high probability**.

iii) Finding the **Good Enough** Solution.

Let us denote the samples in SS by $u_d(j), j = 1, \ldots, s$. Suppose we solve the MDCP(u_d) exactly for each sample in SS, the sample that has the least objective value will be the **good enough** solution that we are looking for as have been concluded by the OO theory [7], [8]. However, solving s ($= 50$) MDCP(u_d), which are nonlinear constrained optimization problems, is too time consuming to meet the requirement of online power system operation. Thus, to resolve this computational difficulty, we employ a two-phase strategy based, again, on the OO theory [7], [8]. Before presenting this two-phase strategy, a tough question that may be raised is what if all s samples in SS are infeasible? We have addressed this question in Appendix C.

Now as shown in Fig. 1, the **basic idea** of our two-phase strategy is to evaluate the s samples in SS very approximately first and order them according to this approximate evaluation. Then, the **actual best alternative** will be contained in the **top few observed good choices**. Thus, in phase 1, we can efficiently solve for **the approximate solutions** of the s MDCP(u_d) and order them based on their corresponding objective values, which represent the estimated performance of the s u_d. Then, in phase 2, we only solve for the exact solutions of the MDCP(u_d) for the few top ranked samples obtained in phase 1, and the one with least objective value is the **good enough** solution that we look for. In the following, we will describe this two-phase approach in detail. In the first phase, we will solve the MDCP(u_d) approximately for a given $u_d(j)$ by solving its quadratic approximate problem as shown in the following:

$$\min_{\Delta x} \frac{1}{2}\Delta x^T[\nabla^2_{xx} f(x^*(u_c^*), u_c^*) + \delta I]\Delta x + [\nabla_x f(x^*(u_c^*), u_c^*) + \nabla^2_{u_c x} f(x^*(u_c^*), u_c^*)(u_c^* - u_d(j))]^T \Delta x$$

subject to

$$g(x^*(u_c^*), u_c^*) + \nabla_{u_c} g^T(x^*(u_c^*), u_c^*)(u_c^* - u_d(j)) + \nabla_x g^T(x^*(u_c^*), u_c^*)\Delta x = \mathbf{0}$$

$$h(x^*(u_c^*)) + \nabla_x h^T(x^*(u_c^*))\Delta x \le \mathbf{0} \qquad (7)$$

where I is an identity matrix, and $\delta > 0$ is a small positive scalar but enough to make the Hessian of (7) positive definite.

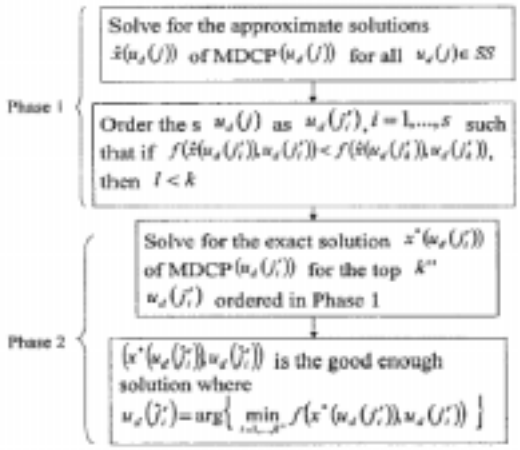

Fig. 1. Two-phase strategy for finding the good enough solution.

We let $\Delta x^*(u_d(j))$ denote the optimal solution of (7), then $\hat{x}(u_d(j)) = x^*(u_c^*) + \Delta x^*(u_d(j))$ is the approximate solution of the optimization problem inside the bracket of (2) for $u_d = u_d(j)$. We have developed an efficient dual-type method called DPPQN in [11], [12], [14], which is especially suited for solving (7) for all samples $u_d(j), j = 1, \ldots, s$. However, in order not to affect the fluency of presentation, we will present this method in Appendix D. Furthermore, this dual-type method can also resolve the computational difficulty caused by the infeasibility of (7) for a given $u_d(j)$. Since the practical objective functions of the OPF such as the total generation cost and the total system losses are mostly convex, the quadratic-approximated MDCP(u_d) shown in (7) should be a good approximate model of the actual MDCP(u_d), thus the order of $u_d(j_i'), i = 1, \ldots, s$ ordered based on the objective value $f(\hat{x}(u_d(j_i')), u_d(j_i'))$ should be closely related to the order of $u_d(j_i''), i = 1, \ldots, s$, ordered based on the objective value $f(x^*(u_d(j_i'')), u_d(j_i''))$.

Therefore, we can estimate the number of top ranked samples in $u_d(j'i)_{i=}1, \ldots, s$ that will consist of the actual top sample $u_d(j_1'')$ based on the alignment probability [10]. That is, to estimate the k^{er} such that $u_d(j_1'') \in \{u_d(j_i'), i = 1, \ldots, k^{er}\}$. Once k^{er} is estimated, we need only to solve the exact solution of MDCP(u_d) for $u_d(j_i'), i = 1, \ldots, k^{er}$. In our problem, $k^{er} = 3$ and the detailed procedures for obtaining k^{er} based on the alignment probability [10] are described in Appendix B.

C. OO Theory-Based Algorithm for Solving the MDCP

Now, we are ready to state our algorithm for solving the MDCP shown in (2) to obtain a **good enough** solution.

Step 1) Solve the CP in (4) using the method proposed in [11], [12] to obtain $(x^*(u_c^*), u_c^*)$.

Step 2) Compute $\Delta f(\Delta u_{cj}^*)$ by (5) for each $j \in \{1, \ldots, n_d\}$, $\Delta u_{cj}^* = u_{cj}^* - \lfloor u_{cj}^* \rfloor$ and $\Delta u_{cj}^* = u_{cj}^* - \lceil u_{cj}^* \rceil$. Pick the l least-value $|\Delta f(\Delta u_{cj}^*)|$, such that $2^{n_d - l} \approx N$, then set u_{dj} to be $\lfloor u_{cj}^* \rfloor$ if $\Delta u_{cj}^* = u_{cj}^* - \lfloor u_{cj}^* \rfloor$, or $\lceil u_{cj}^* \rceil$ if $\Delta u_{cj}^* = u_{cj}^* - \lceil u_{cj}^* \rceil$.

Steps 1 and 2 is how we pick the N samples needed for ordinal optimization.

Step 3) Compute $\Delta f(u_d(i))$ by (6) and calculate $|\Delta f(u_d(i))|$ for all $i = 1,\ldots,N$.

Step 4) Rank all N samples based on their values of $|\Delta f(u_d(i))|$ obtained in Step 3, such that the samples with smaller $|\Delta f(u_d(i))|$ will be ranked higher. Then pick out the top s samples to form SS.

Steps 3 and 4 use a very crude model to form the selected subset (SS) required by OO theory.

Step 5) Solve (7) for all s $u_d(j)$ in SS using the dual projected pseudo Quasi-Newton (DPPQN) method described in Appendix D to obtain the approximate solutions $\hat{x}(u_d(j)), j = 1,\ldots,s$. Order the s $u_d(j)$ based on their approximate objective values $f(\hat{x}(u_d(j)), u_d(j)), j = 1,\ldots,s$, to be $u_d(j_i), i = 1,\ldots,s$, such that if $f(\hat{x}(u_d(j_k)), u_d(j_k)) < f(\hat{x}(u_d(j_l)), u_d(j_l))$, then $k < l$.

Step 6) Find the exact solutions of the MDCP(u_d) in (3) for $u_d(j_i), i = 1,\ldots,k^N$, obtained in Step 5. The one having the least objective value is the **good enough** solution.

Step 5 uses a slightly more accurate but still approximate model to evaluate the few top-ranked samples obtained in Step 4. OO theory then guarantees that the best ranked sample in Step 6 is indeed good enough with high probability.

D. Online Modifications

It is quite possible that before completing the solution process of solving the quadratic programming problems shown in (7) for all samples $u_d(j)$ in SS, we may already obtain an approximate solution $(\hat{x}(u_d(j)), u_d(j))$ whose objective value $f(\hat{x}(u_d(j)), u_d(j))$ is close to $f(x^*(u_c^*), u_c^*)$. Considering the limited computation budget for online optimal power flow application, we should solve for the exact solution of MDCP(u_d) for this sample $u_d(j)$ immediately instead of solving it after all s quadratic approximate solutions are obtained. Consequently, if the objective value of the resulting solution of this sample is close to $f(x^*(u_c^*), u_c^*)$, we have obtained a **good enough** solution. Otherwise, we will go on for the next sample. Such a modification definitely saves computational time. Thus, we can modify steps 5 and 6 of our algorithm presented in previous subsection for online applications as follows.

Step 5M: Set $j = 0$ and set $j = j + 1$; set the values of $\beta(0 < \beta < 1)$ and $\delta_1(> 0)$; set the non-negative integer $m = 0$.

Step 6M: Solve (7) for $u_d(j)$. If $(|f(\hat{x}(u_d(j)), u_d(j)) - f(x^*(u_c^*), u_c^*)|/|f(\hat{x}(u_d(j)), u_d(j))|) < \beta^m\delta_1$, solve the MDCP$(u_d)$ for this $u_d(j)$ to obtain $x^*(u_d(j))$ and go to Step 7M; otherwise, set $j = j + 1$. If $j \le s$, repeat this step; otherwise, go to Step 8M.

Step 7M: If $(|f(x^*(u_d(j)), u_d(j)) - f(x^*(u_c^*), u_c^*)|/|f(x^*(u_d(j)), u_d(j))|) < \varepsilon_1$, where ε_1 is a small positive real number, then $(x^*(u_d(j)), u_d(j))$ is the **good enough** solution and stop; otherwise, set $(x^*(u_d(j)), u_d(j))$ as the temporary **good enough** solution and check whether there are enough computation budget left; if not, stop and output the temporary **good enough** solution; otherwise, go to Step 8M.

Step 8M: If $j > s$, stop and output the temporary good enough solution; otherwise, set $j = j + 1, m = m + 1$ and return to Step 6M.

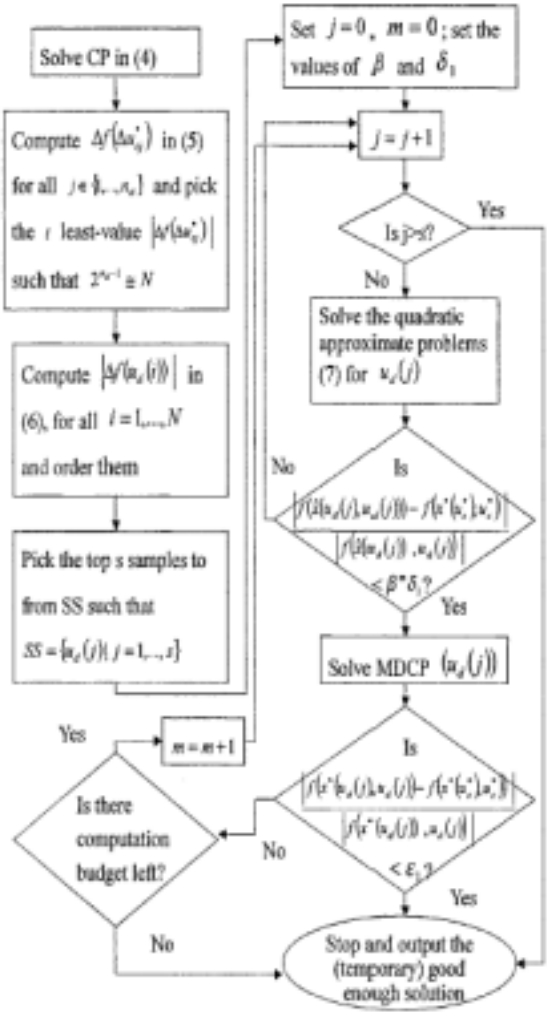

Fig. 2. Flowchart of proposed online algorithm.

With the above modifications, we show the flowchart of our OO theory-based online algorithm in Fig. 2. In addition, a brief illustration about the parameters β and δ_1 is given in the following remark.

Remark 1: The positive real number $\beta^m\delta_1$ in Step 6M is used to measure the closeness between $f(\hat{x}(u_d(j)), u_d(j))$ and $f(x^*(u_c^*), u_c^*)$. Thus, in Step 5M, we can set δ_1 to be a not too small value, say 0.03, so as to obtain a temporary good enough solution $(x^*(u_d(j)), u_d(j))$ in Step 6M fast. Subsequently in Step 7M, if $f(x^*(u_d(j)), u_d(j))$ is close enough to $f(x^*(u_c^*), u_c^*)$, then $(x^*(u_d(j)), u_d(j))$ is the good

enough solution that we are looking for; otherwise, we will increase m by 1 in Step 8M and return to Step 6M. When m increases, $\beta^m\delta_1$ will decrease, because $\beta(<1)$ behaves like a reduction factor. Thus, we can proceed further to obtain a better temporary good enough solution or the good enough solution. We typically set $\beta = 0.9$ in Step 5M.

IV. Test Results

A. Test Systems and Test Cases

We have tested our online algorithm on the OPF problems with discrete control variables of the IEEE 118-bus system and the IEEE 244-bus system [30]; the former consists of 18 generation buses and 179 transmission lines, and the latter consists of 47 generation buses and 445 transmission lines. It should be noted that the values of conductance of the transmission lines in the IEEE 244-bus system are much higher than that of the IEEE 118-bus system on the average. We consider two types of objective function: the minimum total real power generation cost $\Sigma_{G_i} a_i P_{G_i}^2 + b_i P_{G_i} + c_i$ and the minimum system losses $\Sigma_l P_{\mathbf{loss}}(l)$, where P_{G_i} denotes the real power generation of generation bus G_i, a_i, b_i, and c_i are generation cost coefficients, and $P_{\mathbf{loss}}(l)$ denotes the real power loss on transmission line l. For each system and each objective function, we have tested several cases associated with various number of capacitors and various number of transformers. We assume each capacitor is equipped with three capacitor banks, and the capacity of a bank ranges from 14 to 42 MVAR. We assume each transformer tap has 32 discrete steps such that each step is 5/8% of the nominal transformer tap ratio. A summary of all the test cases is described below. Case 1A to Case 4A in Table I and Case 5A to Case 8A in Table II represent the test cases on the IEEE 118-bus system. Case 1A to Case 4A use the total generation cost, while Case 5A to Case 8A use the total system losses as their objective functions. The number of capacitors and the number of transformers assumed in Case 1A–8A are shown in the second and third columns of Tables I and II. Case 1B to Case 4B in Table III and Case 5B to Case 8B in Table IV represent the test cases on the IEEE 244-bus system. Case 1B to Case 4B use the total generation cost, while Case 5B to Case 8B use the total system losses as their objective functions. The corresponding number of capacitors and transformers assumed in these eight cases are also shown in the second and third columns of Tables III and IV. It should be noted that all the tests we have made are carried out in a Pentium IV personal computer.

B. Comparison With the Conventional Approach

Our tests for demonstrating the performance of our online algorithm in comparison with the conventional approach are carried out in the following. We first solve these 16 cases using the **conventional approach**, which solves the CP in (4) for each case first then round the obtained optimal continuous values of the discrete control variables, u_c^*, off to their nearest discrete values. After the values of the discrete control variables are fixed at the nearest u_d, we then solve the MDCP(u_d), and the resulting objective values and the consumed CPU times are shown in the fourth and the seventh column, respectively, of each table. We apply our **online algorithm** presented in Section III-D with $\delta_1 = 0.03, \varepsilon_1 = 0.01$, and $\beta = 0.9$ to solve for a **good enough** solution for each case, and the resulting objective value and the consumed CPU times are shown in the fifth and the eighth column, respectively, of each table. The reduction of the objective values achieved by our online algorithm compared with the conventional approach is given in column 6 of each table. We found that among all of the discrete control variables, the objective value (4) is less sensitive to transformer tap ratio changes than it is to the capacitor bank changes; hence, we set the transformer tap ratios fixed in Step 2 of our online algorithm. This result is consistent with the observation in [15]. The values of the discrete control variables obtained by the conventional approach may be infeasible as appeared in Cases 4A, 1B, and 2B in which infinite objective values are indicated; however, our online algorithm obtains **good enough** solution in all cases we have simulated. This demonstrates the probability of

TABLE I
Comparison of Our Online Algorithm With the Conventional Approach on Cases 1A–4A of the IEEE 118-Bus System Using the Total Generation Cost as the Objective Function

CASE	No. of Cap.	No. of Trans.	Obj. Value ($/hr) Con. Appr.(I)	Obj. Value ($/hr) Our Alg.(II)	Obj. Value Red. $\frac{(I)-(II)}{(II)}\times 100\%$	CPU times (seconds) Con. Appr.	CPU times (seconds) Our Alg.
1A	10	10	40414	39603	17.2%	1.676	2.351
2A	12	12	49500	40306	22.8%	2.541	2.731
3A	15	15	51154	41759	22.5%	2.133	3.312
4A	20	20	∞	41612	∞	-	2.67

TABLE II
Comparison of Our Online Algorithm With the Conventional Approach on Cases 5A–8A of the IEEE 118-Bus System Using the Total System Losses as the Objective Function

CASE	No. of Cap.	No. of Trans.	Obj. Value Con. Appr.(I)	Obj. Value Our Alg.(II)	Obj. Value Red. $\frac{(I)-(II)}{(II)}\times 100\%$	CPU times (seconds) Con. Appr.	CPU times (seconds) Our Alg.
5A	10	10	63.1MW	55.1MW	14.5%	1.236	1.251
6A	12	12	68.2MW	55.2MW	23.6%	1.237	2.732
7A	15	15	82.7MW	50.4MW	64%	1.396	5.078
8A	20	20	82.6MW	48.5MW	70.8%	1.372	1.424

TABLE III
Comparison of Our Online Algorithm With the Conventional Approach on Cases 1B–4B of the IEEE 244-Bus System Using the Total Generation Cost as the Objective Function

CASE	No. of Cap.	No. of Trans.	Obj. Value ($/hr) Con. Appr.(I)	Obj. Value ($/hr) Our Alg.(II)	Obj. Value Red. $\frac{(I)-(II)}{(II)}\times 100\%$	CPU times (seconds) Con. Appr.	CPU times (seconds) Our Alg.
1B	15	15	∞	149708	∞	-	4.379
2B	25	25	∞	166801	∞	-	4.476
3B	30	30	153067	132012	16.0%	4.24	4.31
4B	39	39	150983	132327	14.0%	4.24	4.45

TABLE IV
Comparison of Our Online Algorithm With the Conventional Approach on Cases 5B–8B of the IEEE 244-Bus System Using the Total System Losses as the Objective Function

CASE	No. of Cap.	No. of Trans.	Obj. Value Con. Appr.(I)	Obj. Value Our Alg.(II)	Obj. Value Red. $\frac{(I)-(II)}{(II)}\times 100\%$	CPU times (seconds) Con. Appr.	CPU times (seconds) Our Alg.
5B	15	15	38.4MW	29.5MW	31.0%	5.38	5.82
6B	25	25	42.6MW	36.5MW	16.7%	5.031	3.858
7B	30	30	44.2MW	37.2MW	18.9%	6.379	11.483
8B	38	38	39.8MW	31.7MW	25.8%	5.09	7.220

not obtaining any feasible solution by our algorithm is extremely low comparing with the conventional approach. Furthermore, the CPU times consumed by our online algorithm is slightly larger than that consumed by the conventional approach in most of the cases as can be observed from the last two columns of all tables. Thus, compared with the conventional approach, we observed that: i) our online algorithm is also suitable for real-time application; ii) the improvements on the objective values for either the minimum total generation cost or minimum total system losses are 28% on the average; and iii) we can always obtain a **good enough** solution in contrast to the possible failure of the conventional approach.

C. Comparison With the Competing Methods

Our method deals with the OPF problems with discrete control variables in their exact form and takes the feasibility of OPF into account in contrast to the penalized discretization algorithm [15] and the enhanced genetic algorithm [1]. Thus, we can avoid any possible ambiguity on the feasibility of our solution. The recent methods that treat the discrete control variables and the feasibility of OPF like our algorithm are mean-field theory based annealing algorithm [5], the evolutionary algorithm [17], and the tabu search method [9]. However, these methods are seeking the global optimal solution of the OPF (2) within the huge sample space U_d, which should be very computationally time consuming. For examples, the mean field theory-based annealing algorithm took 10 min to solve the OPF of a 270-bus system as reported in [5], and the tabu search method and evolutionary algorithm took 6 and 27 min, respectively, to solve the OPF of a 24-bus system as described in [9]. In fact, the computational complexity caused by global searching technique is what our OO theory-based online algorithm intend to resolve by seeking a good enough solution with high probability instead.

To demonstrate the merits of our online algorithm, we should compare with the above mentioned methods by simulations. In [5], the mean field theory-based annealing algorithm compared only with the SQP method for a four-bus and 30-bus systems. However, the tabu search method in [9] had compared with and beaten the evolutionary algorithm and the SQP method. Thus, it would be typical to compare our online algorithm with the tabu search method.

The tabu search is an iterative algorithm; it starts from a randomly generated feasible solution and moves to a better solution in the randomly selected neighborhood by the following procedures. Starting from the best solution in the selected neighborhood, if it does not belong to the Tabu List (TL), or in case of being in the TL but passes the aspiration level (AL) test, it will be added to the TL and a move to this solution will be carried out; otherwise, repeat these procedures for the next best solution. During the search process, the best solution is always updated and stored aside until the stopping criterion is satisfied. Details of this method can be found in [9] and [18]. We apply the tabu search method to all of the cases shown in Tables I to IV. Due to the page limitation, we cannot present all of the comparison results. However, we will show some typical cases to demonstrate the efficiency of our online algorithm. Figs. 3–6 describe the simulation results of applying our online algorithm and the tabu search method to Cases 5A, 7A, 5B, and 7B. The objective functions of these four cases are the minimum system losses. In each of the four figures, the point marked by "x" denotes the pair of the objective value of the best solution so far and the corresponding consumed CPU times during the search process of the tabu search method applying to the corresponding case. However, the point marked by "o" in each figure denotes the pair of the objective value of the final solution and the corresponding CPU times obtained

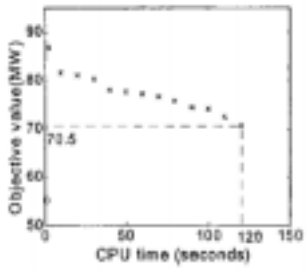

Fig. 3. Comparison of our online algorithm and the tabu search method on Case 5A.

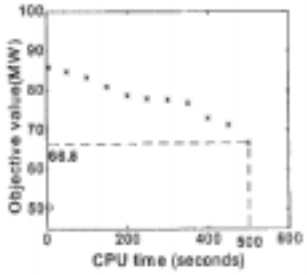

Fig. 4. Comparison of our online algorithm and the tabu search method on Case 7A.

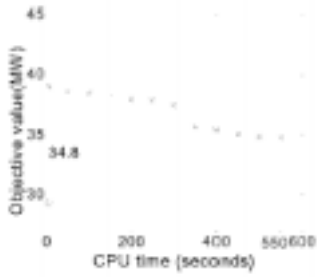

Fig. 5. Comparison of our online algorithm and the tabu search method on Case 5B.

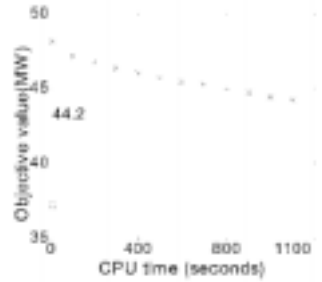

Fig. 6. Comparison of our online algorithm and the tabu search method on Case 7B.

by our online algorithm. We let $(t^{OO}_{(\cdot)}, f^{OO}_{(\cdot)})$ denote the coordinate of point "o", where $(\cdot)$ represents the corresponding case. From column 8 and column 5 of Tables II and IV, we see that $(t^{OO}_{5A}, f^{OO}_{5A}) = (1.251\ \text{s}, 55.1\ \text{MW})$, $(t^{OO}_{7A}, f^{OO}_{7A}) = (5.078\ \text{s}, 30.4\ \text{MW})$, $(t^{OO}_{5B}, f^{OO}_{5B}) = (5.82\ \text{s}, 29.3\ \text{MW})$, and $(t^{OO}_{7B}, f^{OO}_{7B}) = (11.483\ \text{s}, 37.2\ \text{MW})$. We let $(t^{TS}_{(\cdot)}, f^{TS}_{(\cdot)})$ denote the coordinate of the specific point "x", such that $t^{TS}_{(\cdot)} \approx 100 t^{OO}_{(\cdot)}$ in the corresponding case $(\cdot)$. We also mark $(t^{TS}_{(\cdot)}, f^{TS}_{(\cdot)})$ in each figure, that is $(t^{TS}_{5A}, f^{TS}_{5A}) = (120\ \text{s}, 70.5\ \text{MW})$, $(t^{TS}_{7A}, f^{TS}_{7A}) = (500\ \text{s}, 66.8\ \text{MW})$, $(t^{TS}_{5B}, f^{TS}_{5B}) = (550\ \text{s}, 34.8\ \text{MW})$, and $(t^{TS}_{7B}, f^{TS}_{7B}) = (1100\ \text{s}, 44.2\ \text{MW})$. From Figs. 3–6, we see that the improvement of the solution during the search process of the tabu search method is very sluggish. Even when tabu search method consumes $t^{T}_{(\cdot)} (\approx 100 t^{OO}_{(\cdot)})$ CPU times, $f^{TS}_{(\cdot)}$ is still at least 18% more than $f^{OO}_{(\cdot)}$ in each case. These results demonstrate the efficiency of our online algorithm in getting a good enough solution.

V. Concluding Remarks

The OPF problem with discrete control variables is an NP hard problem and has a long history in power system research. In this paper, we have developed an OO theory based online algorithm to deal with it. We treat the discrete control variables in their exact form and take the feasibility of the OPF into account. Our online algorithm can get a good enough solution of the considered problem efficiently to resolve the computational complexity caused by the approach of global searching techniques such as the mean field theory-based annealing algorithm [5], evolutionary algorithm [16], and the tabu search method [9]. We have demonstrated the computational efficiency of our online algorithm and the quality of the obtained solution by comparing with the tabu search method and the conventional approach through simulations.

Above all, our ordinal optimization theory-based algorithm has a very important implication in power system control and management, because most of the power system optimization problems involve discrete control variables; a popular example problem in addition to the current one is the optimal capacitor placement problem.

Appendix A

I. OO Theory [7], [8]

In order to make the computation of the NP hard problem $\min_{u_d \in U_d} f'(u_d)$ tractable, where U_d is a nonstructured huge sample space, the OO theory soften the goal from the **best for sure** to **good enough with high probability** such that a great reduction of the search space can be achieved. To proceed with the search reductions, the OO theory starts from defining a representative set of U_d, denoted by $\mathbf{N}$, which can be obtained by either a uniform selection or a heuristic method. Usually, the size of $\mathbf{N}$, denoted by N, is sufficiently large, say 1000. A **Good Enough Subset** of $\mathbf{N}$, denoted by GS_N, is formed by the top ranked $n\%$, say $n = 3.5$, samples of $\mathbf{N}$. As shown in [13], $\text{GS}_N \subseteq \text{GS}_{U_d}$ with a very high probability 0.991, where GS_{U_d} denotes the **Good Enough Subset** of U_d formed by the top $n_{U_d}\%$, say $n_{U_d} = 5$, samples of U_d. Thus, to seek a **good enough** solution of U_d, it is sufficient to seek a **good enough** solution of $\mathbf{N}$. However, a GS_N is easy to specify but difficult to obtain; for example, in our problem, one needs to solve N (= 1000) $\text{MDCP}(u_d(j))$, that is N nonlinear constrained optimization problems, to determine the GS_N which is, of course, computationally intractable for large power systems. Thus, to reduce the search samples while obtaining members in GS_N, the OO theory advocates the use of a very crude model in evaluating the "goodness" of any proposed solution sample. Using such a crude model one can with manageable effort estimate what are the top $n\%$ of the samples. Call this estimated top $n\%$ the **Selected Subset** (SS). In other words, the SS is determined by first ranking the samples in $\mathbf{N}$ based on an **easily computed crude model** of $f'(u_d)$ noted by $f'(u_d) + \omega$, where ω denotes the modeling noise or error, then choose the top s samples to form SS. The OO theory then provides a **universal alignment probability** between the intersection of the GS_N and the SS [10]. Furthermore, OO theory provides in [10] a formula for obtaining the value s as the value of a six-parameter function $Z(k, g, N, C, \xi(\cdot), P_A)$, the meaning of these parameters is described below. k and g denote the desired number of alignments between SS and GS_N and the size of GS_N, respectively. C denotes a class of ordered performance curve (OPC), a summary of which is presented later in remark 2. The OPC class chosen for the N samples strongly depends on the designer's familiarity of the system, for example, if one is familiar with the structure of the system, he or she may employ a better heuristic method than **uniform selection** to determine the set $\mathbf{N}$ from U_d; $\xi(\cdot)$ represents the noise characteristics of the modeling error ω; a uniform noise density $\text{U}[-\text{W}, \text{W}]$ is assumed, and the magnitude W strongly depends on the crude model $f'(u_d) + \omega$ chosen for determining SS. P_A is the alignment probability and is taken to be 0.95 in most of the applications. Simple expressions for the function $Z(k, g, N, C, \xi(\cdot), P_A)$, based on a Monte Carlo study over numerous OPCs distributed uniformly among the five broadly generic classes are given in [9]. The formula allows us to determine the value s by a simple and direct calculation. Once the value s is determined, we need only to solve the $\text{MDCP}(u_d(j))$ for those s samples of SS, and the resulting k top samples ordered by the objective values of the $\text{MDCP}(u_d(j))$ in SS will be members of GS_N with probability $P_A = 0.95$. In our problem, we need only one **good enough** solution, thus the top ranked sample is what we are looking for.

1) Remark 2: Illustration of the Ordered Performance Curve (OPC) [10]: Consider a standard optimization problem $\min_{u_d \in U_d} f'(u_d)$, where U_d is the sample space, and $f'(\cdot)$ is a performance measure defined on the sample space. The OPC of a collection of ordered samples $u_{d_1}, u_{d_2}, \ldots, u_{d_N}$ selected from U_d is determined by the spread of the order performance $f'_{[1]}, f'_{[2]}, \ldots, f'_{[N]}$, where $f'_{[i]}$ denotes $f'(u_{d_i})$. Without loss of generality, $f'_{[i]}$s can be normalized into the range [0, 1], that is, for $i = 1, \ldots, N$, $y_i = (f'_{[i]} - f'_{[1]})/(f'_{[N]} - f'_{[1]})$. Meanwhile, the ordered samples, spaced equally, are also mapped into the range [0, 1] such that for all $i = 1, \ldots, N$, $x(u_{d_i}) \equiv x_{[i]} = (i-1)/(N-1)$. There are five broad categories of OPC models: i) lots of good samples, ii) lots of intermediate but few good and bad samples, iii) equally distributed good, bad, and intermediate samples, iv)

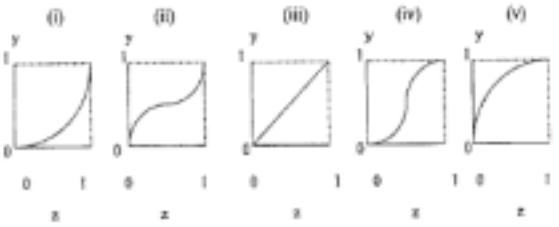

Fig. 7. Graphical expression for the five OPC models.

lots of good and lots of bad samples but few intermediate ones, and v) lots of bad samples. A graphical expression for these five OPC models or types is shown in Fig. 7. To accommodate the above five OPC types and differentiate the curves within one type by using the smallest number of parameters, the inverse mapping of the incomplete Beta function, parameterized by a pair of numbers α and γ, is employed. More precisely, the standardized OPC is determined by a two-parameter smooth curve $\Lambda(z\,|\,\alpha,\gamma) = F^{-1}(z\,|\,\alpha,\gamma) = F(z\,|\,(1/\alpha),(1/\gamma))$, where $F(z\,|\,\cdot,\cdot)$ is the Incomplete Beta function of the two parameters $(\cdot,\cdot)$. In general, $\alpha < 1, \gamma > 1$ corresponds to the OPC of type (i); $\alpha > 1, \gamma > 1$ corresponds to the OPC of type (ii); $\alpha = 1, \gamma = 1$ corresponds to the OPC of type (iii); $\alpha < 1, \gamma < 1$ corresponds to the OPC of type (iv); $\alpha > 1, \gamma < 1$ corresponds to the OPC of type (v).

II. Determination of the Size of Selected Subset (SS) for Our Problem [10]

In our problem, the desired number of alignments between SS and $\mathrm{GS}_{N'}$, k, is 1. The corresponding smallest number of alignments between SS and GS_N, k, to achieve the above desired alignment is also 1 [13] as indicated in Section I.A. Thus, we can determine the value $s = Z(k, g, N, C, \xi(\cdot), P_A)$ by the formula given in [10], in which we take $g = 50, k = 1, N = 1000, P_A = 0.95, C$ is of worst case that is type (v) OPC, and a uniform large noise distribution $\mathrm{U}[-2.5, 2.5]$ for the noise characteristics $\xi(\cdot)$. The value of s for our problem we obtained from the above formula is 50. It should be noted that the value s we obtained based on the above setup is conservative due to the selection of C and $\xi(\cdot)$.

APPENDIX B

To estimate the value of $\mathrm{k}^{N'}$ indicated in Section III.C, we employ the technique of alignment probability [10] as summarized in Appendix A. In the current problem, we apply the following similarities of the terminologies described in Appendix A. We first denote the SS obtained in ii) of the three-stage approach presented in Section III.B as SS^0, then treating this SS^0 as the set $\mathbf{N}$ in OO theory, that is, we set $\mathbf{N} = \mathrm{SS}^0$. Thus, the top ranked sample in $\mathrm{SS}^0 = \{u_d(j_i), i = 1, \ldots, s\}$ ordered based on the objective values of $\mathrm{MDCP}(u_d(j_i))$ will be considered as GS_N. In the first phase of our two-phase strategy presented in Section III.C, we use a good approximate model to determine the estimated good enough subset, that is the SS. Now we should determine the size of the SS to ensure that $\mathrm{GS}_N \subseteq \mathrm{SS}$ with very high probability based on the technique of alignment probability [10]. Since the size of $\mathbf{N}$ here is only around 50, which is far too small to apply the formula provided in [10] as summarized in Appendix A, we have to perform a Monte Carlo study followed from the guidelines given in [10] to estimate the value $\mathrm{k}^{N'}$ (i.e., the size of SS, as described below).

We first place the $s(= 50)$ samples in $N = (\mathrm{SS}^0)$ by equally spacing them in the normalized ordered interval [0, 1]. We studied a total of 49 OPCs distributed uniformly among the five broadly generic types formed from the following parameters $\alpha = 0.2, 0.4, 0.8, 1.0, 2.0, 4.0, 5.0$, and $\gamma = 0.2, 0.4, 0.8, 1.0, 2.0, 4.0, 5.0$, the meaning of the two parameters α and γ are described in Remark 2 of Appendix A. We employ a moderate noise $\xi(\cdot)$ with uniform noise density $\mathrm{U}[-0.5, 0.5]$, because our quadratic approximation for the $\mathrm{MDCP}[u_d(j)]$ given in (7) is in fact a good approximate model from the viewpoint of an OPF problem researcher. As a matter of fact, the range of the noise considered above equals the range of OPC, which can result in (with nonzero probability) swapping the rank of some good enough samples with the worst samples. In all of our Monte-Carlo calculations, we simulate 10 000 realizations of noisy OPCs. For the alignment probability equals 0.99, we found that $\mathrm{k}^{N'} = 3$. In other words, $u_d(j_1^{N'}) \in \{u_d(j_i^f), i = 1, \ldots, \mathrm{k}^{N'}\}$. This implies that after solving the $s(= 50)$ quadratic approximate problems shown in (7), we need only solve $\mathrm{k}^{N'}(= 3)$ exact $\mathrm{MDCP}[u_d(j)]$.

APPENDIX C

We call a sample $u_d(i)$ feasible if the $\mathrm{MDCP}(u_d(i))$ in (3) for the given $u_d(i)$ has a feasible solution; otherwise, we call this $u_d(i)$ infeasible. In addition, we consider the optimal objective value of an infeasible sample as ∞ [3]. Obviously, to obtain the top ranked sample of SS ordered by the objective value of $\mathrm{MDCP}(u_d(i))$, we need to solve the $\mathrm{MDCP}(u_d(i))$ for all samples in SS. However, the samples in SS chosen based on the model (6) do not guarantee to be feasible. Subsequently, we have to answer the question that what if all s samples in SS are infeasible and does one need to search for a feasible solution from $\mathbf{N}$? As a matter of fact, if all s samples of SS are infeasible, then we can be resonably sure (i.e., with probability more than 0.95) the probability that there are feasible samples among the $\mathbf{N}$ is very low as concluded by the method of point estimation of the opinion poll in statistics ([20], Chap. 3 and [22]). This is briefly illustrated below. Suppose the target population of interest consists of N persons and M of them favor a certain proposition. What we should like to know is the proportion $\mathrm{p} = (\mathrm{M}/N)$. We consider the opinion poll as a binomial experiment and treat the sample that favors the proposition as a "success." Then the true success probability p can be estimated by $\hat{p} = k/n$, where $n(\ll N)$ denotes the limited number of trials, and k denotes the observed number of successes. Now a question is how good is the estimate? Let us determine the probability that the estimate $\hat{p}$ does not deviate from the true success probability p by more than some small quantity d. In other words, we want to know the probability-P(d)-that $p - d \le \hat{p} \le p + d$. Substituting $\hat{p}$, P(d) denotes the probability that k lies in the following range $np - nd \le k \le np + nd$. This probability can be evaluated with the help of normal approximation, and $\mathrm{P}(\mathrm{d}) \ge 0.95$ if $n = 1/d^2$.

Now our feasibility test can be viewed as the above opinion poll experiment such that the feasible sample is considered as a success. Using the following similarities, s $\sim$ n and t $\sim$ k, if there are t feasible samples among the s(= 50), then we can be resonably sure that the probability of picking up a feasible one from the N(= 1000) samples is $t/50$ with deviation not more than $1/\sqrt{s}$, which equals 0.1414. Thus, if all s(= 50) samples of SS are infeasible, then we can be reasonably sure the probability that there are feasible samples in the N is 0 with deviation not more than 0.1414. This addresses the question raised in (iii) of Section III.B. In the above argument, we inherently assume that the samples in SS were uniformly selected; however, this is not our case. As a matter of fact, the samples which are more likely to satisfy the constraints $g(x, u_d) = 0$ are more probably selected into SS. This is illustrated below. The criteria that we rank the samples to be selected into SS is according to (6). That is the sample $u_d(i)$ with lower value of $\|\Delta f(u_d(i))\|$ will be ranked higher. Since (6) can be rewritten in the following form $\Delta f(u_d(i)) \cong \lambda^T \Delta g(x^*(u_c^*), u_d^*(i))$, where $\Delta g(x^*(u_c^*), u_d^*(i)) \approx \nabla_{u_c} g(x^*(u_c^*), u_c^*)(u_c^* - u_d(i))$. Thus, if $u_d(i)$ is more likely to be feasible then it will cause lower value of $\|\Delta g(x^*(u_c^*), u_d^*(i))\|$ and will induce lower value of $\|\Delta f(u_d(i))\|$. Consequently, it will be more probably selected into SS.

APPENDIX D

For the sake of simplicity in explaining the dual-type method for solving the quadratic approximate problem (7), we consider the total generation cost as the objective function, and the inequality constraints consist of the generation limits and voltage security constraints only. The same method can be applied to the cases with the total system losses as the objective function. Cases including the coupling thermal limit in the inequality constraints can be similarly treated by taking the modification in [12] into account.

Based on the above assumption, $f(x, u_d)$ is a convex function of x only, thus in the sequel, we use the notation $f(x)$ as the objective function. Since the cost of an individual generator is a quadratic function of the generation, thus $\nabla_x^2 f(x^*(u_c^*))$ is a constant diagonal matrix. Furthermore, since the generation limit and voltage security are decoupled bus-wise, the inequality constraints $h(x^*(u_c^*)) + \nabla_x h^T(x^*(u_c^*))\Delta x \leq 0$ can be decomposed into

$$h_i(x^*(u_c^*)) + \nabla_{x_i} h_i^T(x^*(u_c^*))\Delta x_i \leq 0, i = 1, \ldots, B \quad (8)$$

where B denotes the number of buses in the system; Δx_i and h_i denote the increment of continuous variables and the inequality constraint functions associated with bus i. Thus, (7) can be written in the following form:

$$\begin{aligned} &\min_{\Delta x} \frac{1}{2}\Delta x^T H \Delta x + \tau^T \Delta x \\ \text{subject to} \quad &A^T \Delta x = b(u_d(j)) \\ &C_i^T \Delta x_i \leq d_i, \quad i = 1, \ldots, B \end{aligned} \quad (9)$$

where the constant matrices $H = \nabla_x^2 f(x^*(u_c^*)) + \delta I, A = \nabla_x g(x^*(u_c^*), u_c^*)$, and $C_i = \nabla_{x_i} h_i(x^*(u_c^*))$, the constant vectors $d_i = -h_i(x^*(u_c^*))$ and $\tau = \nabla_x f(x^*(u_c^*))$, and the dependent vector $b(u_d(j)) = -g(x^*(u_c^*), u_c^*) - \nabla_{u_c} g^T(x^*(u_c^*), u_c^*)(u_c^* - u_d(j))$. Clearly, the matrix H is positive definite, and we can assume that A is of full rank, because otherwise we can always delete the redundant equality constraints.

1. DPPQN Method

We will employ the DPPQN method proposed in [11], [12], [14] to solve (9). The DPPQN method solves the following dual problem of (9) instead of solving (9) directly:

$$\max_{\lambda} \phi(\lambda) \quad (10)$$

where the dual function

$$\phi(\lambda) = \min_{\Delta x \in \Omega} \frac{1}{2}\Delta x^T H \Delta x + \lambda^T (A^T \Delta x - b(u_d(j)) \quad (11)$$

and $\Omega = \cup_{i=1}^{B} \Omega_i$, where

$$\Omega_i = \{\Delta x_i \,|\, C_i^T \Delta x_i \leq d_i\} \quad (12)$$

and $\Omega_j \cap \Omega_k = \emptyset$ if $j \neq k$.

The DPPQN method is an iterative method using the following iterations:

$$\lambda(t+1) = \lambda(t) + \tau(t)\Delta\lambda(t) \quad (13)$$

where t is the iteration index, $\tau(t)$ is a positive step-size determined by an Armijo-type rule [11], [12], [14], and the increment $\Delta\lambda(t)$ is computed by

$$\Phi\Delta\lambda(t) + \nabla_\lambda \phi(\lambda(t)) = 0 \quad (14)$$

where the matrix

$$\Phi = -A^T H^{-1} A \quad (15)$$

is a negative definite matrix, which represents the Hessian of $\phi(\lambda)$, however, not considering the constraints $\Delta x \in \Omega$ as will be shown later in (17). The gradient of $\phi(\lambda)$ at $\lambda(t)$ can be computed by

$$\nabla_\lambda \phi(\lambda(t)) = A^T \Delta\hat{x} - b(u_d(j)) \quad (16)$$

where $\Delta\hat{x}$ is the optimal solution of the optimization problem on the right-hand side (RHS) of (11), [14].

As shown in [11] and [12], the optimal solution for the optimization problem on the RHS of (11), $\Delta\hat{x}$, can be obtained using a two-phase approach. The first phase is to solve the following unconstrained optimization problem analytically:

$$\min \frac{1}{2}\Delta x^T H \Delta x + \lambda^T (A^T \Delta x - b(u_d(j)) \quad (17)$$

which is the optimization problem on the RHS of (11) without considering the constraints $\Delta x \in \Omega$. Let $\Delta\bar{x}$ be the optimal solution of (17), then

$$\Delta\bar{x} = -H^{-1}(A\lambda). \quad (18)$$

The second phase is to project the $\Delta\bar{x}$ onto Ω, that is projecting $\Delta\bar{x}_i$ onto Ω_i, for $i = 1, \ldots, B$, and the resulting projection is the optimal solution of the optimization problem on the RHS of (11), $\Delta\hat{x}$.

1) Summary of the DPPQN Method: Starting from a $\lambda(t)$, we can use the two-phase method mentioned above to solve for

$\Delta\tilde{x}$, the optimal solution of the optimization problem on the RHS of (11). Once $\Delta\tilde{x}$ is obtained, we can compute $\nabla_\lambda\phi(\lambda(t))$ by (16). Associated with the matrix Φ expressed in (15), we can solve for $\Delta\lambda(t)$ from (14). We then update $\lambda(t+1)$ by (13) using an Armijo-type step-size $\tau(t)$ and proceed with the next iteration.

II. Computational Efficiency

We see that the matrix Φ in (15) is sparse, because H is diagonal, and the nonzero entries of A represent the structure of the connectivity of the power system, which is sparse. Thus, we can obtain $\Delta\lambda(t)$ at each iteration t by solving the sparse linear (14) using sparse matrix technique. However, the most distinguished point for applying the DPPQN method here is that the matrix Φ is a **constant sparse matrix for all samples** $u_s(j)$ *in* SS. This implies that the LU factorization and the associated memory management for the nonzero entries for the sparse matrix Φ is done once and for all. This, of course, will save tremendous computational-time in solving all the s dual problems (10).

III. Convergence and Solution

The Φ in (15) is a negative definite matrix, because H is positive definite and A is of full rank. Thus, $\Delta\lambda(t)$ obtained from (14) will be an ascent direction of the dual function (11). With the Armijo-type step-size $\tau(t)$ [11], [12], [14], convergence of the DPPQN method had been shown in [11], [12], and [14]. Let λ^* denote the optimal solution of the dual problem (10), and $\Delta\tilde{x}(\lambda^*)$ denotes the optimal solution of the optimization problem on the RHS of (11) when $\lambda = \lambda^*$. Since our primal problem (9) is a quadratic programming problem with a strictly convex objective function, by strong duality theorem [3], the $\Delta\tilde{x}(\lambda^*)$ is the solution of (9). Furthermore, if the dual function $\phi(\lambda)$ tends to approach ∞ during the solution process, we can conclude that $u_s(j)$ is infeasible for (7) with the objective value ∞.

REFERENCES

[1] A. Bakirtzis, P. Biskas, C. Zoumas, and V. Petridis, "Optimal power flow by enhanced genetic algorithm," *IEEE Trans. Power Syst.*, vol. 17, pp. 229–236, May 2002.

[2] A. Bakirtzis and A. Meliopoulos, "Incorporation of switching operation in power system corrective control computations," *IEEE Trans. Power Syst.*, vol. 2, pp. 669–676, Aug. 1987.

[3] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[4] R. Burchet, H. Happ, and D. Vierath, "Quadratically convergent optimal power flow," *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 3267–3275, Nov. 1985.

[5] L. Chen, H. Suzuki, and K. Katou, "Mean field theory for optimal power flow," *IEEE Trans. Power Syst.*, vol. 12, pp. 1481–1486, Nov. 1997.

[6] T. Giras and N. Talukar, "Quasi-Newton method for optimal power flows," *Int. J. Electric Power Energy Syst.*, vol. 3, no. 2, pp. 59–64, Apr. 1981.

[7] Y. C. Ho, *Soft Optimization for Hard Problem*. Cambridge, MA: Harvard Univ., 1996. Lecture Notes.

[8] Y. C. Ho, C. C. Cassandras, C. H. Chen, and L. Dai, "Ordinal optimization and simulation," *J. Oper. Res. Soc.*, vol. 21, pp. 490–500, 2000.

[9] T. Kulworawanichpong and S. Sujitjorn, "Optimal power flow using tabu search," *IEEE Power Eng. Rev.*, pp. 37–40, June 2002.

[10] T. W. E. Lau and Y. C. Ho, "Universal alignment probability and subset selection for ordinal optimization," *J. Optim. Theory Appl.*, vol. 39, no. 3, June 1997.

[11] C. Lin and S. Lin, "A new dual-type method used in solving optimal power flow problems," *IEEE Trans. Power Syst.*, vol. 12, pp. 1667–1675, Nov. 1997.

[12] ——, "Improvements on the dual-type method used in solving optimal power flow problems," *IEEE Trans. Power Syst.*, vol. 17, pp. 315–323, May 2002.

[13] S. Lin and Y. C. Ho, "Universal alignment probability revisited," *J. Optim. Theory Appl.*, pp. 399–407, May 2002.

[14] S. Lin and C. Lin, "A computationally efficient method for nonlinear multicommodity network flow problems," *Networks*, pp. 225–244, July 1997.

[15] W. Lin, A. Papalexopoulos, and W. Tinney, "Discrete shunt controls in a Newton optimal power flow," *IEEE Trans. Power Syst.*, vol. 7, pp. 1509–1520, Nov. 1992.

[16] D. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.

[17] J. T. Ma and L. L. Lai, "Evolutionary programming approach to reactive power planning," *Proc. Inst. Elect. Eng.*, vol. 143, Apr. 1996.

[18] A. Mantawy, Y. Abdel-Magid, and S. Selim, "Unit commitment by tabu search," *Proc. Inst. Elect. Eng.-Gen. Transm. Distr.*, vol. 145, no. 1, pp. 56–64, 1998.

[19] A. Monticelli and W. Liu, "Adaptive movement penalty method for the Newton optimal power flow," *IEEE Trans. Power Syst.*, vol. 7, pp. 334–340, Feb. 1992.

[20] G. E. Noether, *Introduction to Statistics—The Nonparametric Way*. New York, NY: Springer-Verlag, 1991.

[21] B. Stott, J. Marinho, and O. Alsac, "Review of linear programming applied to power system rescheduling," *Proc. IEEE Power Ind. Comput. Applicat. Conf.*, pp. 142–154, May 1979.

[22] B. Stott and J. Marinho, "Linear programming for power system network security applications," *IEEE Trans. Power App. Syst.*, vol. PAS-98, pp. 847–848, May/June 1979.

[23] B. Stott, O. Alsac, and A. Monticelli, "Security analysis and optimization," *Proc. IEEE*, vol. 75, pp. 1623–1664, Dec. 1987.

[24] D. Sun, B. Ashley, B. Brewer, A. Hughes, and W. Tinney, "Optimal power flow by Newton approach," *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 2864–2880, Oct. 1984.

[25] D. Sun, T. Hu, G. Lin, C. Lin, and C. Chen, "Experiences with implementing optimal power flow for reactive scheduling in the Taiwan power systems," *IEEE Trans. Power Syst.*, vol. 3, pp. 1193–1200, Aug. 1988.

[26] S. Talukdar and T. Giras, "A fast and robust variable metric method for optimum power flows," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 415–420, Feb. 1982.

[27] W. Tinney, J. Bright, K. Demaree, and B. Hughes, "Some deficiencies in optimal power flow," *Proc. IEEE PICA Conf.*, pp. 164–169, May 1987.

[28] G. Torres and V. Quintana, "On a nonlinear multiple-centrality-corrections interior-point method for optimal power flows," *IEEE Trans. Power Syst.*, vol. 16, pp. 222–228, May 2001.

[29] Y. Wu, "Fuzzy second correction on complementary condition for optimal power flows," *IEEE Trans. Power Syst.*, vol. 16, pp. 360–366, Aug. 2001.

[30] Y. Wu, A. Debs, and R. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Trans. Power Syst.*, vol. 9, pp. 876–883, May 1994.

Shin-Yeu Lin is currently a Professor in the Department of Electric and Control Engineering at National Chiao Tung University, Hsinchu, Taiwan, R.O.C. His research interests include optimal power flow, ordinal optimization theory, and applications.

Yu-Chi Ho received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA.

Since 1969, he has been Gordon McKay Professor of Engineering and Applied Mathematics at Harvard University. In 1988, he was appointed to the T. Jefferson Coolidge Chair in Applied Mathematics and Gordon McKay Professor of Systems Engineering at Harvard and as Visiting Professor to the Cockrell Family Regent's Chair in Engineering at the University of Texas, Austin. His research interests include differential games, information structure, multiperson decision analysis, to incentive control, and since 1983, exclusively in discrete event dynamic systems, ordinal optimization, perturbation analysis, and manufacturing automation.

Ch'i-Hsin Lin is currently an Associate Professor in Electronics Engineering Department at the Kao Yuan Institute of Technology, Kaoshiung Taiwan, R.O.C. His research interests include large scale power systems and ordinal optimization theory and applications.