New concepts in neighborhood search for permutation optimization problems

Wojciech Bożejko[1] and Mieczysław Wodecki[2]

[1] Institute of Engineering Cybernetics, Wrocław University of Technology
Janiszewskiego 11-17, 50-372 Wrocław, Poland
email: wbo@ict.pwr.wroc.pl
[2] Institute of Computer Science, University of Wrocław
Przesmyckiego 20, 51-151 Wrocław, Poland
email: mwd@ii.uni.wroc.pl

1 Introduction

For many strongly NP-hard combinatorial optimization problems a natural representation of a solution is a permutation. Practical approaches to solve such problems are local search algorithms based on the neighborhood's search. Well chosen neighborhood is one of the key elements, which affects efficiency of this method. Classic neighborhoods have polynomial number of elements and they are generated by single moves. Neighborhoods with an exponential number of elements have been successfully applied for a few years. We introduce the neighborhood (and its properties – neighborhood graph, diameter, etc.), which has application in the best local search algorithms. This neighborhood belongs to very large scale neighborhood class (VLSN) and it is generated by swap multimoves. We present a theorem which states that any permutation (solution) can be transformed into any other permutation by execution at most two swap multimoves (that is the diameter of the neighborhood graph equals 2). The second theorem of this paper states that the problem of determining an optimal swap multimove in the neighborhood is NP-hard. Applying this neighborhood (and the algorithms of its exploration) to local search algorithms allows us to obtain the best known results for the most difficult scheduling problems considered in literature (single machine total weighted tardiness problem, flow shop and job shop problems).

In this paper we propose a neighborhood generated by a composition of all the swap moves, where supports of the permutation connected with these moves are disjoint - that is different swaps of a multiswap does not change the same elements of the permutation (they are commutative). Such a neighborhood is very large (has an exponential number of elements). Determining its optimal element is an NP-hard problem.

One of the characteristics of the neighborhood structure is the diameter of the corresponding graph. We will also prove, that the diameter of the multiswap neighborhood graph is 2. Applying this neighborhood to local search algorithms allows us to obtain the best known results for the most difficult scheduling

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 363–366. ISBN 80-210-3726-1.

problems considered in literature (e.g. single machine total weighted tardiness problem [2], flow shop [6] and job shop problems [7]).

2 Swap multimoves

We consider a combinatorial optimization problem with a permutational representation of a solution. Let $\mathcal{I} = \{1, 2, \ldots, n\}$ be a set of n elements (enumerated by numbers from 1 to n). By $S(n)$ we mean a set of all permutations of the set $\mathcal{I}$.

Permutation $\sigma \in S(n)$ is called *involution* [8], if it is inverse to itself, i.e. $\sigma^2 = \epsilon$, where ϵ is the identity permutation. It is simple to see, that if $\sigma \in S(n)$ is an involution, then an inverse permutation σ^{-1} is an involution too.

Fact 1 *Every involution is a composition of pair-independent (with disjoint supports) transpositions.*

Conclusion 1. If $\pi \in S(n)$ and m is a swap move, then executing of this move generates a permutation $\beta = m(\pi)$. If the move m swaps an element $\pi(i)$ with $\pi(j)$, therefore it is easy to see, that $\beta = m(\pi) = \pi\alpha$, where $\alpha = \begin{pmatrix} 1\ 2 \ldots i-1\ i\ i+1 \ldots j-1\ j\ j+1 \ldots n \\ 1\ 2 \ldots i-1\ j\ i+1 \ldots j-1\ i\ j+1 \ldots n \end{pmatrix}$ is a transposition. So the move m can be identified with a transposition α. Therefore Fact 1 follows, that an involution is a composition of swap moves (and we call it a multiswap).

Theorem 1 [3] *For any permutations $\pi, \delta \in S(n)$ there exists involutions $\alpha, \beta \in S(n)$ such, that $\pi\alpha\beta = \delta$.*

Conclusion 2. From Theorem 1 and Fact 1 it follows that for any permutation $\delta \in S(n)$ there exist involutions $\alpha = \alpha_1\alpha_2 \ldots \alpha_l$ and $\beta = \beta_1\beta_2 \ldots \beta_l$, where $\alpha_s\beta_s$ $(s = 1, 2, ..., l)$ are independent cycles and such that $\delta = \alpha\beta$. The method of constructing of both involutions is precisely described in the proof of the theorem.

Lemma 1 *The diameter of the neighborhood graph corresponding to multimove swap neighborhood is 2.*

Proof. We consider any two permutations $\pi, \delta \in S(n)$ - nodes of the neighborhood graph $G = (V, A)$. From Theorem 1 it follows, that $\pi\alpha\beta = \delta$, where permutations α, β are involutions. Because $\pi\alpha \in \mathcal{N}(\pi)$, then $(\pi, \pi\alpha) \in A$ (that is the length between them is 1). Similarly, $\delta \in \mathcal{N}(\pi\alpha)$, therefore $(\pi\alpha, \delta) \in A$. It follows that the diameter of the multiswap neighborhood graph is 2. ■

Remark 1. In one of the advanced heuristic methods – path relinking – for two permutations π and δ a permutation γ, lying on a path between π and δ is constructed. Because $\pi\alpha\beta = \delta$, where α, β are involutions, so permutation $\gamma = \pi\alpha$, which is generated from π by an involution (multimove) α, lies on the path between π and δ. It is possible to construct such an involution α that permutation γ is in the required range of distance to π. In particular construction

of involutions α, β (see Conclusion 2) can be successfully applied to diversify the calculations, as a fully deterministic method. From construction involution with many transpositions it follows that permutation $\pi\alpha$ is at a long distance (counted as a number of swap moves) from π.

Remark 2. For any permutation $\pi \in S(n)$ the set of all permutations $S(n)$ can be divided into three subsets (orbits): a) $\{\pi\}$, b) $\{\pi\alpha : \alpha \in S(n)$ and α is an involution $\}$, c) $\{\pi\alpha\beta : \alpha, \beta \in S(n)$ and α, β are involutions $\}$. The dynasearch neighborhood from the paper [4] is generated by single multimoves and these neighborhoods are subsets of the set defined in point b).

Remark 3. The neighborhood based on swap multimoves is of very large-scale (the number of its elements is asymptotically $\frac{1}{\sqrt{2}} \left(\frac{n}{e}\right)^{\frac{n}{2}} e^{\sqrt{n}-\frac{1}{4}}$, see [8]). We will prove that the problem of finding the optimal multiswap in the multimove swap neighborhood is equivalent to finding an optimal traveling salesman path in a certain graph.

Theorem 2 [3] *The problem of determining an optimal multiswap in the multimove swap neighborhood is NP-hard.*

Remark 4. In the works [6],[7] and [2] multimoves are applied to intensify and diversify the calculations. Because these moves are compositions of independent moves, therefore one can estimate the goal function of the permutation generated by these moves with good precision.

3 Application: The permutation flow shop scheduling

Garey, Johnson and Seti [5] show that for three machine flow shop problem $(F|3|C_{\max})$ is strongly NP-hard. The best available branch and bound algorithms are those of Lageweg, Lenstra and Rinnooy Kan [9]. Their performance is not entirely satisfactory however, as they experience difficulty in solving instances with 20 jobs and 5 machines. Various local search methods are available for the permutation flow shop problem. A very fast tabu search algorithms is proposed by Grabowski, Wodecki [6]. Multimoves are applied to diversify calculations in this algorithm. We have checked how they influence computations time and values of solutions.

The implementation of the tabu search algorithm TSGW [6] was tested on benchmark instances taken from the OR-library [1] and compared with reference results from this library. For comparison, the results of the TSGW algorithm and the TSGWnoMM algorithm (i.e. TSGW without multimoves) are presented in Table 3. The results are shown of two groups of columns: one is for the TSGW algorithm, the other is for the TSGWnoMM algorithm. Time in seconds (CPU) and percentage relative deviation (PRD) to the reference solutions are presented in Table 1.

Table 1. Quality test results.

	TSGW		TSGWnoMM	
$n\|m$	CPU	PRD	CPU	PRD
20|5	0.0	0.00	0.0	0.02
20|10	0.3	0.03	0.3	0.08
20|20	0.6	0.07	0.6	0.11
50|5	0.4	-0.08	0.4	0.05
50|10	0.9	-0.29	0.8	0.17
50|20	2.3	0.13	2.2	0.26
100|5	0.6	-0.03	0.6	0.14
100|10	1.4	-0.21	1.3	0.32
100|20	5.0	-0.68	4.5	0.19
200|10	4.4	-0.19	4.1	0.06
200|20	8.5	-1.12	7.9	-0.04
500|20	11.2	-0.81	10.7	0.23
all	2.96	-0.26	2.62	0.12

On the basis of results presented in Table 1 we can say, that computations times are almost the same. However definitely different are average relative deviations to the reference solutions. Average relative percentage deviation (PRD) for the TSGW (with multimoves) is -0.26, however for the TSGWnoMM (without multimoves) average PRD is 0.12. Applying of the multimoves cause about 3 times decreasing of the average PRD.

References

1. Beasley J.E., OR-Library: distributing test problems by electronic mail, Journal of the Operational Research Society 41, 1990, 1069-1072.
2. Bożejko W., M.Wodecki, Parallel algorithm for some single machine scheduling problems, Automatics vol. **134** 2002 81-90.
3. Bożejko W., M.Wodecki, On the theoretical properties of swap multimoves, Operations Research Letters, Elsevier (in press).
4. Congram, R.K., C.N. Potts, S.L. Van de Velde, An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem, INFORMS Journal on Computing vol. **14** No. 1 2002 52-67.
5. Garey M.R., D.S. Johnson, R. Seti, The complexity of flowshop and jonshop scheduling, Mathematics of Operations Research, 1, 1976, 117-129.
6. Grabowski J., M. Wodecki, A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion, Computers and Operations Research **31** 2004 1891-1909.
7. Grabowski J., M. Wodecki, A very fast tabu search algorithm for the job shop problem. In: Rego C., Alidaee B., editors. Adaptive memory and evolution; tabu search and scatter search, Dordrecht, Kluwer Academic Publishers 2005.
8. Knuth D.E., The art of computer programming, Vol. 3., second edition, Addison Wesley Longman, Inc. 1998.
9. Lageweg B.J., J.K., Lenstra, A.H.G. Rinnooy Kan, A General Bouding Scheme for the Permutation Flow-Schop Problem, Operations Research, 26, 1978, 53-67.