

GLSDock – Drug Design Using Guided Local Search

Sally Chen Woon Peh¹(✉) and Jer Lang Hong²

¹ School of Biosciences, Taylor's University, Subang Jaya, Malaysia
chenwoonpeh@yahoo.com

² School of Computing and IT, Taylor's University, Subang Jaya, Malaysia
jerlang.hong@taylors.edu.my

Abstract. The guided local search method has been successfully applied to a significant number of NP-hard optimization problems, producing results of similar caliber, if not better, compared to those obtained from algorithms specially designed for each singular optimization problem. Ranging from the familiar TSP and QAP to general function optimization problems, GLS sits atop many well-known algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA) and Tabu Search (TS). With lesser parameters to adjust to, GLS is relatively simple to implement and apply in many problems. This paper focuses on the potential applications of GLS in ligand docking problems via drug design. Over the years, computer aided drug design (CADD) has spearheaded the drug design process, whereby much focus has been trained on efficient searching in de novo drug design. Previous and ongoing approaches of meta heuristic methods such as GA, SA & TS have proven feasible, but not without problems. Inspired by the huge success of Guided Local Search (GLS) in solving optimization problems, we incorporated it into the drug design problem in protein ligand docking and have found it to be effective.

Keywords: Bioinformatics · Guided local search · Protein ligand docking · Drug design · Meta-heuristics

1 Introduction

Drugs only bind to receptors with complementary structures. It is why ligand docking is highly dependent on how well a ligand would effectively fit and bind to any binding site. In computer aided drug design, 'docking' is referred to as the prediction of the binding mode between ligand and a known or estimated structure of a receptor (Fig. 1). This process is affected by many variables [1], one of which is the equilibrium state of the ligand-receptor structure. The free energy of binding molecules must be low. Hence, a scoring function or an objective function is required as a representation of the binding energy as an approach to predict ligand docking. Example approaches are Genetic Algorithm, Simulated Annealing and Tabu Search.

Ligand binding can be predicted through 3 aspects: functional site detection, functional site similarity [2] and molecular docking [3]. For the location of binding sites, some approaches focus on geometric matches (i.e. SURFNET; LIGSITE,

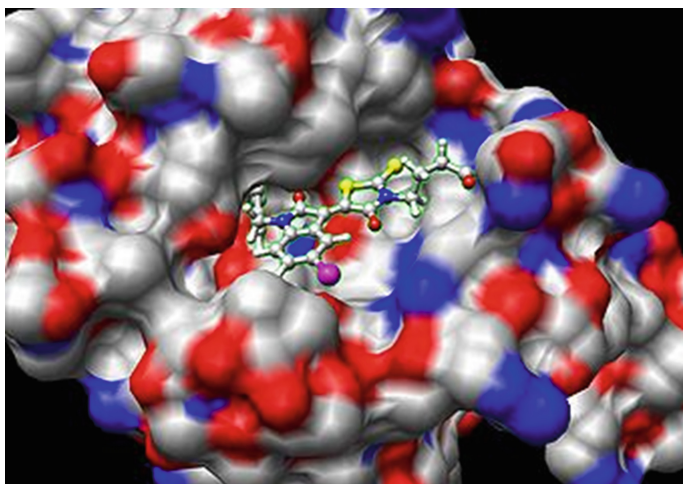


Fig. 1. Protein ligand docking

POCKET, APROPOS, CAST, vdw-fft, Drugsite) [4]. Many are evolutionary techniques [5] and are highly effective, because there is an abundance of information on structural and sequence related data. Knowledge on evolutionary of protein and ligand structures can be obtained from chemical databases such as the Protein Data Bank, Cambridge Structural Database, sc-PDB and more.

Prediction is further made complicated with internal flexibility within ligands, receptors and solvent molecules [6]. This is why large amounts of translational and external rotational degrees of freedom are associated with these molecules, contributing to computational complexity. Ligands are non-static in nature with frequent changes in geometry. They may form similar 3D conformation upon binding to receptors, and active conformation is hard to predict. Current computational processing process is not sufficient for thorough combing of vast search spaces for optimal solutions. This issue is becoming increasingly important due to progressive availability of protein structures from high throughput protein purification, X-ray crystallography and Nuclear Magnetic Resonance spectroscopy.

To create efficient docking softwares, both the scoring function and optimization algorithms are key concerns that are crucial. In this paper, we propose a meta-heuristic technique to simulate the molecular docking process. Looking at past successes of meta-heuristic applications in protein ligand docking based approaches, we are positive that GLS, a top optimization metaheuristic solution is able to provide new approaches to the ligand docking problem. The following sections describe various related meta-heuristics, GLS, applications of GLS and the methodology of GLS in ligand docking followed by our conclusion.

2 Related Work

2.1 Meta Heuristic Techniques

Metaheuristic techniques are non-exact optimization techniques that are applied to problems classified as NP-hard or NP Complete (nondeterministic polynomial time complete) as well as combinatorial optimisation problems. They aim to guide underlying heuristics in solving specific problems in terms of improved efficiency. Although great at providing solutions on large scale instances, metaheuristics share a common problem of being computationally expensive and often requires fine-tuning to solve a particular problem. Sections 3 and 4 will cover the capabilities of GLS in overcoming this computational hurdle.

2.1.1 Genetic Algorithm

A widely used heuristic that imitates Darwin's evolution and natural selection, Genetic Algorithms remain the most popular metaheuristic to date. GA was invented by John Holland who works on the principle of evolution whereby human genes that have undergone crossover and mutation subsequently produce future generations with better traits. This technique includes operators such as crossovers to pass on preferred qualities in an individual to offsprings. During rare circumstances, mutation is applied to promote diversity within the new population by changing variables in the parent solutions. After passing the evaluation process, these new individuals rejoin the population to collectively change the composition of the population.

Jones et al. [7] formulated GOLD dock with implementation of GA and have achieved 71 % of a success rate in docking 100 ligands with root mean square of less than 2 Å. It has also been proven that both the traditional GA and Lamarkian GA [8] far exceed the capabilities of the Monte Carlo SA in handling more degrees of freedom. Additionally, it has been used in a wide variety of computational algorithms for molecular recognition such as DIVALI and SSGA. More reviews on GA applications in molecular docking have been compiled by Willette [9].

GA has been used alongside GLS in the hybrid metaheuristic Guided Genetic Algorithm (GGA) [10]. When the current solution can no longer be further improved, GLS modifies the objective function by means of penalties for GA to use in future generations. Needless to say, these penalties greatly affect mutation and crossover operators in GA to introduce in high numbers, specific characteristics to a population. This allows for increased focus in its search for optimum solutions.

2.1.2 Simulated Annealing

Simulated Annealing [11] works are based on the non-linear cooling process of glass where various stages of changes precede the formation of a crystal. It is assumed that different rates of cooling lead to different formations of glass. The system is first 'melted' at a high effective temperature. Annealing is the lowering of the temperature via slow stages to ensure the system 'freezes' so that there are no other changes. This technique performs a check at every interval of cooling where if the energy is lower, the step of cooling is accepted. However, controlled uphill steps are incorporated in the

search for generalized iterative improvement in solutions. SA has been incorporated in many other heuristics to optimize problems [12].

In ligand docking, this Metropolis Method (SA) is used for positional search and refinement of the ligand-receptor binding conformation. MCDOCK [13] utilises SA to widely sample the binding site for discovery of the local minimum, and has managed to predict fairly accurately (rms of 0.25–1.84 Å) the binding modes of ligands while taking full flexibility into account. Glide [14] incorporates SA in the optimization of its scoring function and has proven to be superior to Dock, GOLD and FlexX.

2.1.3 Tabu Search

Tabu Search (TS) is a heuristic which works on the principle of searching for the optimal solution without revisiting previous search spaces by checking on neighboring areas with its memory storage. This storage takes the form of a tabu list that contains previously visited solutions. This technique improves on the local search (LS) method by relaxing the few basic rules: a worsening move will only be accepted if there are no other improving moves. In addition to that, Tabu Search encourages exploration of otherwise difficult areas by prohibiting moves that lead to previous search areas.

Unlike other metaheuristics, TS is less common in its application in ligand docking. Baxter et al. pioneered the usage of TS in ligand docking with PRO_LEADS [15]. Found to be on par with SA, GA and Evolutionary Programming (EP), TS was used to sample the conformational space. Although slower in computational time, TS was found to be superior to both FlexX and GOLD in terms of prediction rates and estimation of binding affinities.

Both TS and GLS possess the same function of guiding LS out of the local optima. The penalties in GLS are similar to the restrictions implemented in TS. Like GLS, an initial solution is first created at random to be the current solution, and if it is deemed the best after evaluation, it will be added into the Tabu List. Eventually the list will be updated with increasingly better solutions for more intensive search. Because too many penalties would misguide LS, a limited number of penalties are used in later versions of GLS. Old penalties are overwritten and replaced to help the algorithm escape from local minima. A combined approach of TS and GLS was applied to the service network design problem (SNDP) with commendable results [16].

2.1.4 Ant Colony System

The Ant Colony System mimics the behavior of ants in a colony when searching for the shortest route to and back from a feeding source. Every time an ant finds a food source, it will drop pheromones along its path to the food source and back to its nest as a guide for fellow ants. Each ant chooses with high probability to (or not to) follow the original pathway, thus reinforcing the pathway. Upon finding a shorter route, new pheromone trails are laid out. Eventually, the new passageway will be the current preferred route from increased pheromones dropping, due to relatively faster travel. It is assumed that the constant updating of path will eventually lead to an optimal solution. This population based approach is easily applied without major modifications to the Job-shop scheduling problem (SP), Quadratic assignment problems (QAP) and similar versions of the Travelling Salesman Problem (TSP). Tabu lists have also been incorporated in ants as memory storage for improved search results [17].

3 GLS

Many real life problems cannot be realistically solved by a complete search. Similarly, many NP-hard problems are unlikely to be solved via constructive search due to impossibly high computational demand. This led to the development of local search or heuristic methods as an alternative. LS searches in the space of (mostly) randomly generated candidate solutions, then moves on to better ‘neighbors’ until there are no better solutions than the current option. Solutions of LS can be obtained in a relatively short amount of time, but tend to be trapped in local optima. To continue looking for the global minima, GLS guides LS out of the minima via the implementation of penalties. Certain features are banned so that the algorithm focuses its search in more promising areas. Voudouris [18] surmised the similarity of GLS to the Frequency Based Memory approaches in Tabu Search, in which GLS additionally considers both the structural solution and feedback from local optimization heuristic.

To apply this penalty based metaheuristic algorithm, a candidate solution is first defined as a set of features. Each feature will be associated with a cost and penalty, which are the terms and coefficients from the objective function. When the algorithm settles in local optima, the cost function is augmented by accumulating penalties on selected features. These penalty terms are dynamically manipulated throughout the course of the search to steer the heuristic towards more viable solutions. Naturally, the overall cost will be greatly affected by costly features. This way, GLS is able to focus and distribute its searching efforts into more promising areas besides avoiding the accumulation of unnecessary workforce in any one region of the search space. After iteration of the improvement process, the improved solution is assigned as the current best solution until stopping criterion is met.

Similar to Tabu Search, GLS utilises knowledge gained from the previous searches to guide heuristics out of local optima. It provides more flexibility for exploitation of features of a problem in terms of associated costs. Therefore, GLS is able to converge to a high quality solution much more quickly than other metaheuristics such as Tabu Search or Simulated Annealing.

As mentioned, GLS requires definition of problem features. Voudouris and Tsang (1997) detailed the equations reiterated below. Penalties are first initialized at 0 to be subsequently increased as LS reaches a local optimum. GLS defines a function h that will replace g , the objective function which maps candidate solutions s to numerical values. λ is the parameter to the GLS algorithm; i refers to range of features; p_i is the penalty for feature i and I_i indicates the exhibition of feature i (present or absent of exhibition):

$$h(s) = g(s) + \lambda \times \sum [p_i \times I_i(s)] \quad (1)$$

$$I_i(s) = 1 \text{ if } s \text{ exhibits feature } i; 0 \text{ otherwise} \quad (2)$$

To take into account the current penalty value of features, $util_i$ (utility of penalizing feature) is defined as follows, with c_i the cost and P_i the current penalty value of feature i .

$$util_i(s^*) = I_i(s^*) \times \frac{c_i}{1 + P_i} \quad (3)$$

From the equation above, it is clear that the higher the cost (the greater the c_i), the greater the utility of penalizing it. Conversely, the higher the number of times penalized (the greater the P_i), the lower the utility of penalizing it. The idea is to consider both cost and current penalty in the bid to focus searches in more promising search spaces defined by lower cost features i.e. ‘good features’ (Fig. 2).

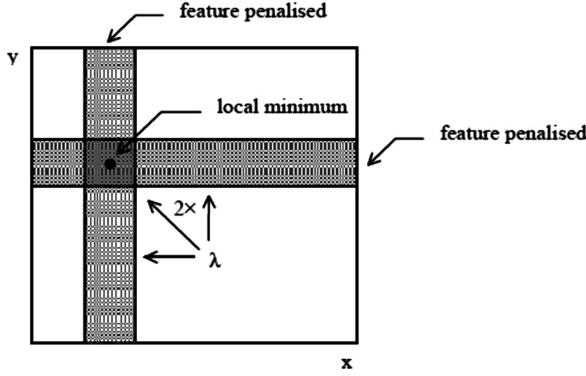


Fig. 2. Penalising features of local minimum to change cost [17]

3.1 Successful Applications of GLS

GLS has been successfully applied in a wide range of well-known NP-hard problems with world class results. In the Radio Link Frequency Assignment Problem, the combination of GGA and GLS hold some of the best results in the CALMA set of benchmark problems, which is the most widely used. GLS also achieved outstanding results in the vehicle routing problem (VRP), another NP-hard problem. In the workforce scheduling problem (WSP), GLS and Fast local search (FLS) achieved the best results in this benchmark problem of minimizing a function involving many variables in the assignment of technicians to jobs [18]. GLS approach in the Optimal Communication Spanning Tree (OCST) problem has also outperformed EA approaches equipped with state-of-the-art search operators [19]. One other highly notable example is the combination of GLS+FLS+2Opt [20] in outperforming Lin-Kernighan algorithm (LK), the specialized algorithm for TSP in average. This GLS combination also achieved superior results compared to SA, TS and GA.

Other than that, general function optimisation problems too benefited from GLS which managed to find consistent solutions in a landscape where local sub-optimal are in abundance. The results show that GLS is capable of defining artificial features for problems without obvious features from the objective function. The Team Orienteering Problem (TOP) found the implementation of GLS to have improved computational time with similar caliber results from other heuristics [21]. Furthermore, GLSSAT (extension of GLS) managed to produce results produced by WalkSAT in the

Max-SAT problem. Once again, a general-purpose algorithm accomplished results comparable to that achieved by a specialized solution to a problem.

In addition to that, GGA obtained results comparable to those produced by GA but with improved robustness in the Generalized Assignment Problem in which agents are assigned to jobs. In 2012, Barbuchu introduced agent based GLS [20], which found satisfactory results in the VRP.

4 Motivation

GLS approaches have proven to be effective in a wide variety of optimization problems. It is within our interest to incorporate this meta-heuristic into the ligand docking problem as we believe that it can be a leading tool in the prediction of binding modes and affinities. We are of the opinion that GLS can bring improvements to solving the docking problem. First, it has been proven that this method could yield better results in Travelling Salesman Problem (TSP) than the traditional State of the Art Genetic Algorithm method. Also, it is fast in converging to the optimal solution within a reasonable time frame. Lastly, implementation of GLS for problem solving has proven to be fast and effective besides having a simple and clear approach. As a meta-heuristic, GLS has proven to have wide applications in varied problems with its ability in creating artificial solution features without prior conclusion of present features from the objective function.

5 Proposed Solution

To implement Guided Local Search efficiently, we implement a local search to drive the search to a local optima (Fig. 3). We use the same scoring function to evaluate the binding energy of docking methods. We represent the docking process by constructing the ligand and protein structure using 3D modelling. Once the 3D modelling of the protein and ligand structure is constructed, we then represent the various atoms and bonds between molecule structures.

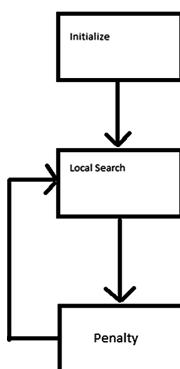


Fig. 3. Flow chart of guided local search

To test our search technique on the docking problem, we divided the docking process into two stages, the first being the orientation of the ligand and the second being the energy generated after binding of ligand and their respective protein structure. Our local search technique alters the orientation of molecules based on two factors, the rotational and degree of freedom. It uses a greedy search that attempts to bind the ligand to the protein structure through best fit. Even the ligand conformation to the protein matched, the respective molecules pairing may eventually lead to non-binding scenario. For such a case, we calculate the binding energy generated after the molecules successfully paired. Every time a ligand is bind to the protein structure, their energy is evaluated. We use scoring function to evaluate the binding energy. Binding energy is evaluated based on the following scoring function:

$$\Delta G_{\text{bind}} = \Delta G_{\text{vdw}} + \Delta G_{\text{H-bond}} + \Delta G_{\text{hydrophobic}} + \Delta G_{\text{rotor}}$$

Where ΔG_{vdw} is the van der Waal energy, $\Delta G_{\text{H-bond}}$ is the hydrogen bonding, $\Delta G_{\text{hydrophobic}}$ is the hydrophobic bonding, and ΔG_{rotor} is the rotatable bonds.

If the energy reduces a lot, the molecule is altered slightly on the assumption that it has nearly reach local optima. Alteration is made on the molecule rotational degree of freedom by twisting the orientation of some of the atoms. Otherwise, the molecule is altered by a large portion so that the energy is reduced drastically. If the energy increases after binding, the new ligand structure is not accepted on the assumption that this step leads to a poorer solution. Instead, we use the same old ligand structure and perform the same step again.

Under certain circumstances, a molecule is altered randomly even the energy decreases. We perform this step as a penalty function, on the assumption that it could lead the solution out of the local optima. On the other hand, if the solution converges to local optima, we modify the scoring function by adding “extra values” to some of the scoring functions. The penalty we imposed on the scoring function may eventually lead to a better solution. The addition of “extra values” is made based on the following criteria:

Van der Waals Energy

Extra values are added to all the atoms which bind between protein and ligand

Hydrogen Bonding Energy

Extra values are added to the H-H binding between protein and ligand

Rotatable Bonds

Extra values are added to the atom or molecule involved in the rotation

Hydrophobic Interaction Energy

Extra values are added to every hydrophobic interactions

The addition of “extra values” is based on the overall effect the various binding energies generated, for example, we would anticipate that hydrophobic interactions generates the least energy among all the binding interactions, hence a smaller penalty values is assigned.

6 Experimental Evaluation

We evaluate our proposed method with the state of the art system AutoDock. We are of the opinion that AutoDock is the most suitable benchmark for our evaluation since AutoDock uses meta-heuristic techniques as part of its operation and it is freely available. We measure the success rate of docking by calculating the binding energy of the protein ligand docking. We test our method with respect to that of AutoDock on protein databank (PDB database), where we randomly choose 50 random samples comprising simple and complex molecules. In addition to that, we measure the execution time for convergence to the optimal solution in the docking process (Fig. 4).

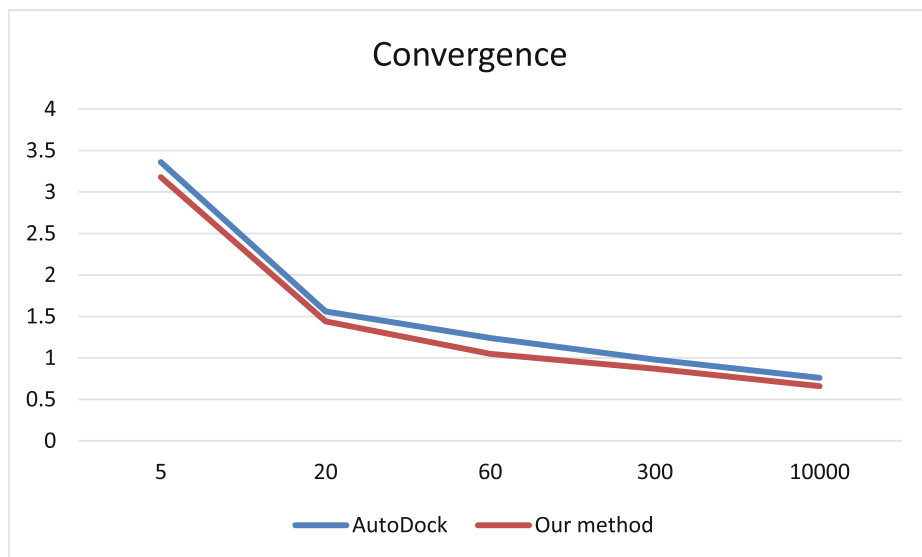


Fig. 4. Convergence graph (Color figure online)

As shown in Table 1, our docking method is more efficient in docking the protein molecular structure. This is due to the fact that we propose a penalty function, where our method impose penalty to the solution randomly, on the assumption that it could jump out from the local optima. On the other hand, we adopt a fast heuristic technique

Table 1. The average binding energy with respect to execution time

Time (s)	AutoDock	Our method
5	3.36	3.18
20	1.56	1.44
60	1.24	1.05
300	0.98	0.87
10000	0.76	0.66

which could converge to the suboptimal solution within reasonable time frame with high accuracy. We believe that our novel proposed method is useful for future protein ligand docking process.

7 Conclusions

It is clear from the results that GLS is a resourceful tool for efficient and effective ligand docking. Similar to past success in application to combinatorial problems, GLS once again proved that it is a potential meta-heuristic for yet another optimization problem. Meta-heuristic techniques are proven to solve many combinatorial optimization problems that involves computationally expensive processing power. We are of the opinion that further refinements on GLS will enable it to stand atop other heuristics in the field of ligand docking.

Acknowledgement. This work is carried out within the framework of a research grant funded by Ministry of Higher Education (MOHE) Fundamental Research Grant Scheme (Project Code: FRGS/1/2014/ICT1/TAYLOR/03/1).

References

1. Huang, S.Y., Zou, X.: Advances and challenges in protein-ligand docking. *Int. J. Mol. Sci.* **11**, 3016–3034 (2010)
2. Schmitt, S., Kuhn, D., Klebe, G.: A new method to detect related function among proteins independent of sequence and fold homology. *J. Mol. Biol.* **323**, 387–406 (2002)
3. Taylor, R.D., Jewsbury, P.J., Essex, J.W.: A review of protein small-molecule docking methods. *J. Comput. Aided Mol. Des.* **16**, 151–166 (2002)
4. An, J., Totrov, M., Abagyan, R.: Comprehensive identification of “druggable” protein ligand binding sites. *Genome Inform.* **15**, 31–41 (2004)
5. Devi, R.V., Sathya, S.S., Coumar, M.S.: *Appl. Soft Comput.* **27**, 543–552 (2015)
6. Cecchini, M., Kolb, P., Majeux, N., et al.: Automated docking of highly flexible ligands by genetic algorithms: a critical assessment. *J. Comput. Chem.* **25**, 412–422 (2003)
7. Jones, G., Willett, P., Glen, R.C., et al.: Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **267**, 727–748 (1997)
8. Ross, B.J.: A Lamarckian evolution strategy for genetic algorithms. In: *The Practical Handbook of Genetic Algorithms*, p. 16 (1999)
9. Willett, P.: Genetic algorithms in molecular recognition and design. *Trends Biotechnol.* **13**, 516–521 (1995)
10. López-Camacho, E., García Godoy, M.J., García-Nieto, J., et al.: Solving molecular flexible docking problems with metaheuristics: a comparative study. *Appl. Soft Comput.* **28**, 379–393 (2015). doi:[10.1016/j.asoc.2014.10.049](https://doi.org/10.1016/j.asoc.2014.10.049)
11. Bertsimas, D., Tsitsiklis, J.: Simulated annealing. *Stat. Sci.* **8**, 10–15 (1993)
12. Suman, B., Kumar, P.: A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Oper. Res. Soc.* **10**, 1143 (2006)
13. Liu, M., Wang, S.: MCDOCK: a Monte Carlo simulation approach to the molecular docking problem. *J. Comput. Aided Mol. Des.* **13**, 435–451 (1999)

14. Friesner, R.A., Banks, J.L., Murphy, R.B., et al.: Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* **47**, 1739–1749 (2004). doi:[10.1021/jm0306430](https://doi.org/10.1021/jm0306430)
15. Baxter, C.A., Murray, C.W., Clark, D.E., et al.: Flexible docking using tabu search and an empirical estimate of binding affinity. *Proteins* **15**, 367–382 (1998)
16. Bai, R., Kendall, G., Qu, R., Atkin, J.A.D.: Tabu assisted guided local search approaches for freight service network design. *Inf. Sci. (NY)* **189**, 266–281 (2012). doi:[10.1016/j.ins.2011.11.028](https://doi.org/10.1016/j.ins.2011.11.028)
17. Dorigo, M., Maniezzo, V., Colomi, A.: The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **26**(1), 1–13 (1996)
18. Voudouris, C.: Guided local search: an illustrative example in function optimisation. *BT Technol. J.* **16**, 46–50 (1998)
19. Tsang, E., Voudouris, C.: Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Oper. Res. Lett.* **20**, 119–127 (1997). doi:[10.1016/S0167-6377\(96\)00042-9](https://doi.org/10.1016/S0167-6377(96)00042-9)
20. Barbucha, D.: Agent-based guided local search. *Expert Syst. Appl.* **39**, 12032–12045 (2012). doi:[10.1016/j.eswa.2012.03.074](https://doi.org/10.1016/j.eswa.2012.03.074)
21. Vansteenwegen, P., Souffriau, W., Berghe, G.V., Van Oudheusden, D.: A guided local search metaheuristic for the team orienteering problem. *Eur. J. Oper. Res.* **196**, 118–127 (2009). doi:[10.1016/j.ejor.2008.02.037](https://doi.org/10.1016/j.ejor.2008.02.037)

Computational Science and Its Applications - ICCSA
2016

16th International Conference, Beijing, China, July 4-7,
2016, Proceedings, Part III

Gervasi, O.; Murgante, B.; Misra, S.; Rocha, A.M.A.C.;
Torre, C.M.; Tanir, D.; Apduhan, B.O.; Stankova, E.;
Wang, S. (Eds.)

2016, XXVII, 574 p. 230 illus., Softcover

ISBN: 978-3-319-42110-0