3rd Symposium on non globular proteins August 28 – September 1, 2007 – Košice - Slovakia

Partitioning biological networks in k-plex subnetworks with maximum edge weights

Milana Grbić[1], Aleksandar Kartelj[2], Dragan Matić[1], Vladimir Filipović[2]

[1] *Faculty of Natural Science and Mathematics, University of Banja Luka*
[2] *Faculty of Mathematics, University of Belgrade*

Introduction

Partitioning large biological networks into smaller clusters can help in discovering new properties and functionalities of a particular structure. In this work we deal with the partitioning of the edge-weighted networks into *k*-plex components, where a set of some *n* vertices in a network is a *k*-plex if the degree of each vertex in the induced subnetwork is at least *n*-*k*. The aim of the maximum edge-weight *k*-plex partitioning problem (Max-EkPP) is to find the *k*-plex partition with the maximum total weight.

Biological justification for using k-plex partitiong instead of clique partitiong

Biological networks are often sparse, therefore many potentially useful information about the interference of biological objects can be neglected if such a restrictive condition is involved. Among many other clique based approaches used for example in protein threading analysis (Akutsu, T. et al. 2006) and other relaxation approaches which can be found in literature (Pattillo, J. et al. 2013) in Max-EkPP, partitioning follows the principle that the objects in each partition are still highly connected in a particular way, but not so restrictively to form a clique. By relaxing the cliques into more sparse graphs, biological objects are grouped in semantically of functionally logical groups which we call k-plexes, having in mind that the total sum of weights in all partitions still should be as great as possible.

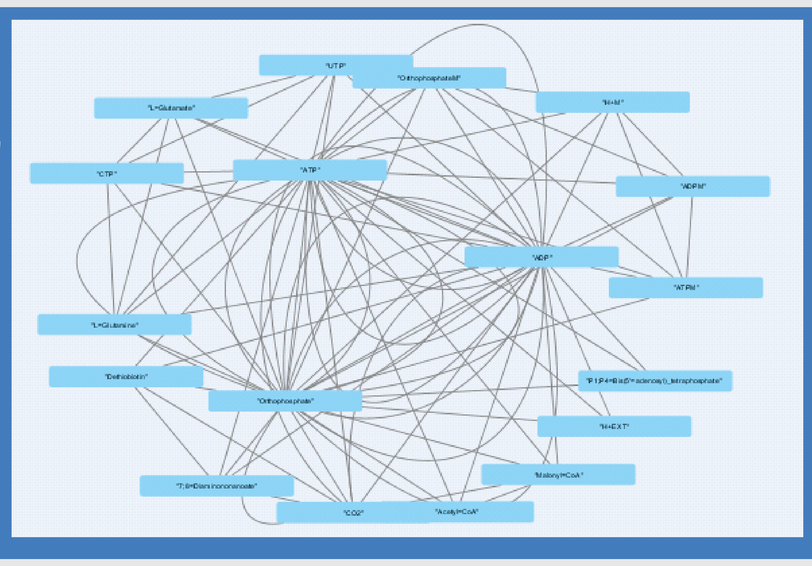
Our approach

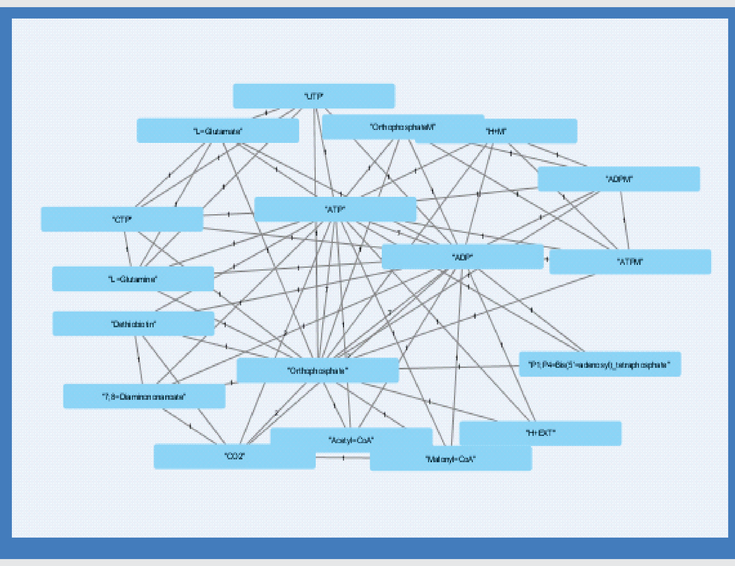
We partition some sparse metabolic networks into *k*-plexes by a local search heuristic method. Following the approach presented in (Martins, 2016.), we apply our algorithm on the metabolic networks. While the exact algorithm based on the integer linear programming (ILP) formulation proposed in (Martins, 2016.) can solve only some of such large-scaled networks, our algorithm can successfully find the solution of any network, including very large- scaled ones. Comparing to the ILP from (Martins, 2016.), our approach succeeds to find most of known optimal solutions, also finding high quality solutions for those instances for which the optimal solution is not known.

Building the network from metabolic reactions

In the example, the network is modeled from the list of 1393 metabolic reactions, taken from (Förster, J. et al., 2003). Metabolites are represented as nodes and two metabolites are adjacent if they figure in at least one common reaction. In a dual approach, metabolic reactions are considered as nodes, while two reactions are adjacent if they share at least one same metabolite.

"ADP" + "ATPM" + "Orthophosphate" -> "ADPM" + "ATP" + "H+M" + "OrthophosphateM"
"Acetyl=CoA" + "ATP" + "CO2" -> "ADP" + "Malonyl=CoA" + "Orthophosphate"
"ADP" + "ATP" -> "Orthophosphate" + "P1,P4=Bis(5'=adenosyl)_tetraphosphate"
"ADP" + "Dethiobiotin" + "Orthophosphate" -> "7,8=Diaminononanoate" + "ATP" + "CO2"
"ATP" -> "ADP" + "H+EXT" + "Orthophosphate"
"ATP" -> "ADP" + "Orthophosphate"
"ATP" + "L=Glutamine" + "UTP" -> "ADP" + "CTP" + "L=Glutamate" + "Orthophosphate"
...
Example:
Metabolites "ADP" and "ATP" participate in 7 common reactions
Metabolites "Orthophosphate" and "CO2“ participate in 2 common reactions





Description of the heuristic algorithm

The considered problem is NP hard, so exact algorithms can not be used for solving large-scaled networks in a reasonable time. Therefore, the usage of heuristic methods is justified. In our approach, we use a local search technique for improving the solution quality. The algorithm starts with an initial solution in which each vertex belongs to its own component - singleton. In each iteration, the algorithm is trying to regroup the vertices consecutively swapping the components of pairs of vertices. Since infeasible partition can appear by a swap, the algorithm also takes into the consideration the degrees of vertices in each partition, favoring feasible solutions vs infeasible ones. The algorithm stops if one of two conditions is satisfied: total number of iterations is reached (in our case 25000), or the best found solution is not improved in 10000 iterations.

Graphical representation

In order to further investigate the results obtained by our heuristic technique, we decided to graphically interpret the results by using a well known software platform for visualizing molecular interaction networks Cytoscape (Shannon, P. et al. 2003) . We use this software to present our results in more suitable way, enabling biologists to better understand the relations between the considered objects. Therefore, we adopted our results in a way that can be easily interpreted by the Cytoscape software,getting more suitable representation of the obtained results.

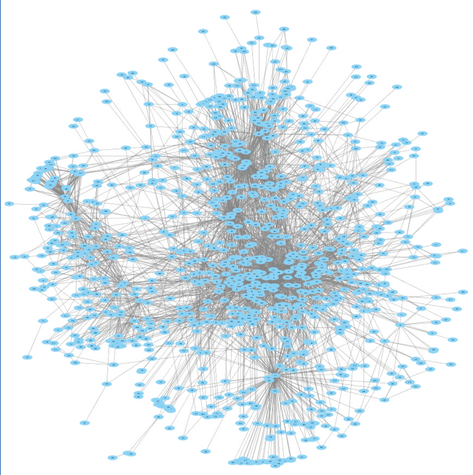
Parallel edges are merged into a single edge, with the weight equals to the total number of common reactions

Instance	|V|	|E|	Density	*k*	OPT/Best	ILP	Our approach Best	Our approach Avg	Our approach Time [s]	#Components	Largest Component
SC-NIP-m-t1	991	4161	0.00848	1	1866	OPT	**1862**	**1857.3**	**1698.42**	504	8
				2	1714	N/A	**best**	**1292.6**	**131.52**	476	7
				3	1395	N/A	**best**	**1074.7**	**155.04**	567	8
SC-NIP-m-t3	177	269	0.01727	1	910	OPT	**OPT**	**910**	**22.15**	110	5
				2	1021	OPT	**OPT**	**1013**	**14.765**	99	7
				3	1139	N/A	**best**	**1046.7**	**14.43**	90	8
SC-NIP-m-t5	75	84	0.03027	1	723	OPT	**OPT**	**723**	**2.95**	43	4
				2	801	OPT	**OPT**	**801**	**2.45**	39	4
				3	887	OPT	**OPT**	**869.1**	**3.22**	36	5

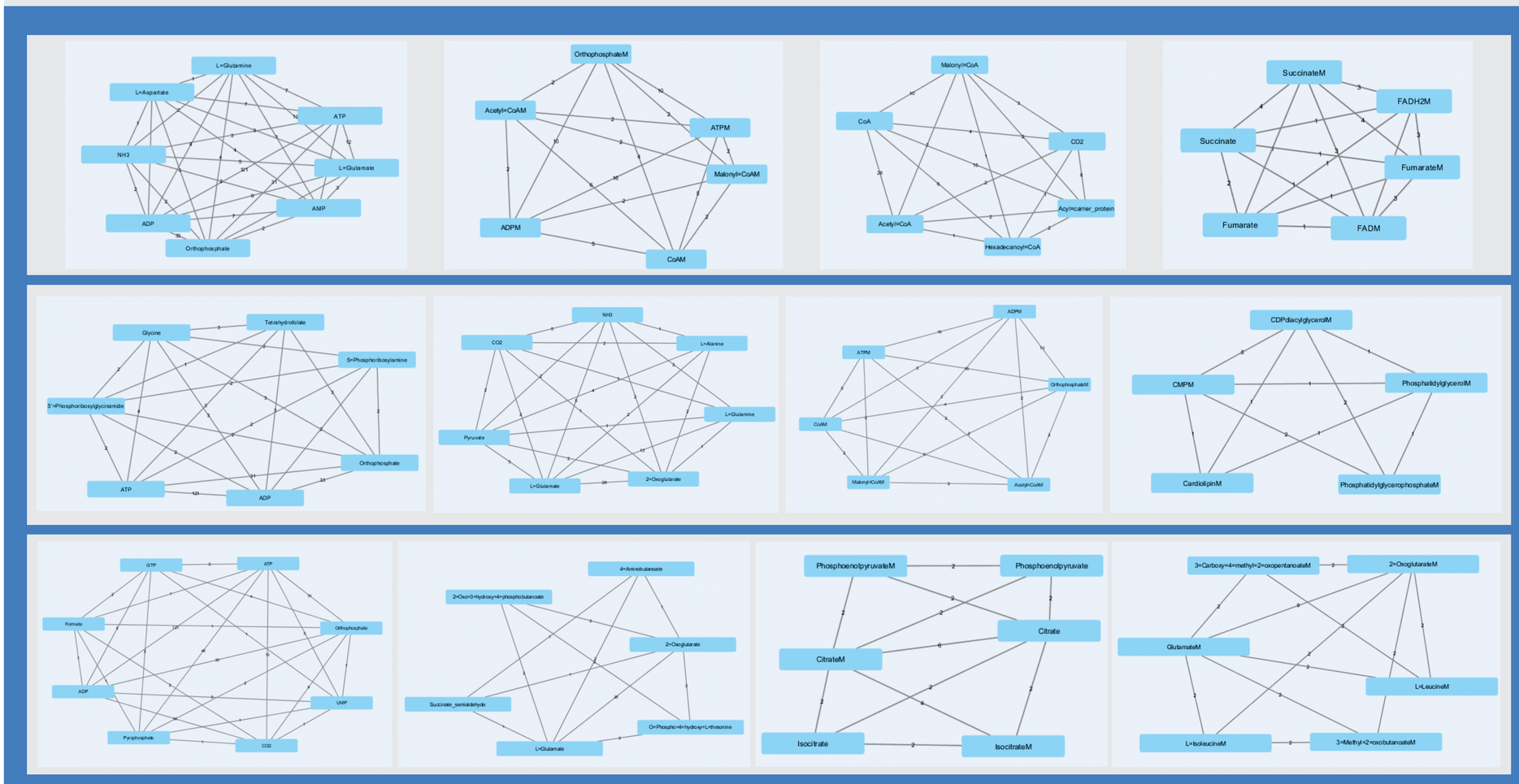
Experimental results

All experiments are performed on the Intel i7-4770 CPU @3.40 GHz with 8 GB RAM and Windows 7 operating system. For each execution, only one thread/processor is used. The algorithm is implemented in C programming language and compiled with Visual Studio 2013 compiler.
For the purpose of this poster presentation, we applied our algorithm on three metabolic instances (Martins, 2016.), namely on SC-NIP-m-t1, SC-NIP-m-t3 and SC-NIP-m-t5, for $k = 1,2,3$. Recall that if $k=1$, the partitions are cliques. For each instance, 10 independents runs were performed.

k=1



k=2



k=3

Conclusions

The maximum edge-weight *k*-plex partitioning problem is of a great interest from both theoretical and practical points of view. This work contains some preliminary results obtained by applying the local search heuristic method on some metabolite networks.
In near future, we plan to improve our algorithm to be more robust and accurate and to apply it to other real life biological and artificial instances from literature.
Obtained results can be used for discovering new biologically important information.

References

Martins, P. (2016). Modeling the maximum edge-weight k-plex partitioning problem. *arXiv preprint arXiv:1612.06243*.
Förster, J. et al. (2003). "Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network." *Genome research* 13.2, 244-253.
Akutsu, T. et al. (2006). "Dynamic programming and clique based approaches for protein threading with profiles and constraints." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 89.5, 1215-1222.
Pattillo, J. et al. (2013). "On clique relaxation models in network analysis." *European Journal of Operational Research* 226.1 9-18.
Shannon, P. et al. (2003) "Cytoscape: a software environment for integrated models of biomolecular interaction networks." *Genome research* 13.11 2498-2504.