Table of Contents

Content Page

# Updated and Expanded Description of the Problem to be Solve

# Background Information and Research

* General:

Heuristics is easy to implement and maintain, and it can be used to solve the problems such as data-mining, cutting, packing, and scheduling; it also have been used in the areas such as anti-virus technology. But the individual heuristics not always works well, sometimes the can make serious error.

On the other hand, there are many problem-solving computer algorithms. But they are too knowledge-intensive or too hard to implement for most potential users.

Hence there is a conflict between using cheap but unstable heuristics and using the complicated domain methodologies.

A possible solution is Hyper-heuristics. 🡪

* Analysis of Initial Brief （what is Hyper-heuristics）:

Hyper-heuristics are heuristics to choose (low-level) heuristics.

(\* A randomly generated initial candidate solution is improved iteratively using a set of low level heuristics in a simple choice hyper-heuristic framework. At each iteration, a given solution passes through two successive stages: heuristic selection and acceptance. The heuristic selection mechanism chooses and applies a low level heuristic to a candidate solution producing a new solution. Then, the acceptance strategy decides whether to continue the search process using the new solution or the one at hand. (Ozcan’s description) \*)

Hyper-heuristics usually aim at reducing the amount of specific knowledge in the researching.

Can save cost for one research

Nowadays the interest for hyper-heuristics is growing.

So it is meaningful and necessary to develop software to introduce the Hyper-heuristic to the potential users 🡪 our project aims at the visualization tools for Hyper-heuristics.

* Research on Existing Systems

There are some visualization tools for Hyper-heuristics (such as) but their functions are quite limited as well as the interface.

* market research
* Results of technical research into suitable platforms, tools, technologies, algorithms, data structures, etc.

# Requirements Speciﬁcation

Do some surveys e.g. (The Questionnaire)

-Functional Requirements:

-Non- functional Requirements:

\*use some diagram (use-case, dataflow, sequence….)

# Initial Design

Key implementation

Conceptual Design

software process model?( Prototyping, Waterfall; Iterative development; Component-based software engineering)

Technical Design

- UML Diagrams

- User Interface

# Discussion of Problems

Technical Issues :

Non-technical problems (such as Management Issues):

# Time Plan

PERT Diagram

Normal Gantt Chart

# Appendix



Time plan

Assign our work and role

Research current solutions

Write Requirement Specification

Write design document

Prototype user interface

Write prototype GUI implement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks | Predecessor | Duration | Exective | Notes |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |